МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра информационно-аналитических систем безопасности

имени Л.С. Берштейна

**ОТЧЕТ**

**Лабораторная работа №5**

**Основы алгоритмизации и программирования**

**«Работа со структурами»**

**Вариант №8**

Выполнил:

студент гр. КТбо1-8

Кябишева А.З.

«\_\_\_» 2017 г.

Проверил:

Профессор кафедры ИАСБ

Беляков Станислав Леонидович

«\_\_\_» 2017 г.

Таганрог – 2017

Оглавление

[Цели работы 3](#_Toc499755574)

[Алгоритм выполнения работы 4](#_Toc499755575)

[Создание алгоритма 4](#_Toc499755576)

[Программный код 5](#_Toc499755577)

[Тестирование программы 7](#_Toc499755578)

[Вывод 8](#_Toc499755579)

# Цели работы

Цель данной лабораторной работы ознакомить студента с понятием структуры данных, научить его правильно использовать ее и работать с ней.

Задание. Написать программу, которая записывает с клавиатуры в файл структуру

согласно выданному варианту задания. В качестве разделителя полей структуры

использовать символ табуляции. В программе реализовать:

а) дополнение существующего массива структур новыми структурами;

б) поиск структуры с заданным значением выбранного элемента;

в) вывод на экран содержимого массива структур;

г) упорядочение массива структур по заданному полю (элементу), например

государство по численности.

Вариант задания: 9. «Владелец автомобиля»: имя, номер автомобиля, номер техпаспорта, дата рождения, номер телефона, отделение регистрации ГИБДД.

# Алгоритм выполнения работы

Когда программа запускается, сразу выводит на экран меню выбора: 1) Внести нового владельца в файл, 2) Поиск нового владельца в файле 3) Вывод всех владельцев на экран 4) Сортировка владельцев в файле 5) Завершение работы программы.

Рассмотрим работу каждого пункта программы на основе двух введенных фильмов.

1) Выбор цифры 1 перебрасывает нас в функцию additionStucture, где пользователь циклично вводит структуру, которая затем записывается в файл.

2) Выбор цифры 3 перебрасывает нас в функцию outputStructure. Функция циклично выводит по 1 структуре из файла.

3) Выбор цифры 4 перебрасывает нас в функцию streamliningStructure. Сначала функция записывает все структуры из файла в один массив структур. Затем, применяя пузырьковую сортировку, выводи отсортированный массив структур на экран(сортирует по возрастнию).

4) Выбор цифры 5 перебрасывает нас обратно в функцию main, где программа просто завершается.

# Программныйкод

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <stdlib.h>

#define MAX\_LENGTH\_NAME 30

#define MAX\_LENGTH\_AUTONUM 10

#define MAX\_LENGTH\_NUMTECHPASS 30

#define MAX\_LENGTH\_BIRTHDATE 10

#define MAX\_LENGTH\_PHONENUMBER 18

#define MAX\_LENGTH\_OTDGIBDD 70

#define MAX\_LENGTH MAX\_LENGTH\_NAME +\

MAX\_LENGTH\_AUTONUM +\

MAX\_LENGTH\_NUMTECHPASS +\

MAX\_LENGTH\_BIRTHDATE +\

MAX\_LENGTH\_PHONENUMBER +\

MAX\_LENGTH\_OTDGIBDD + 5

struct AutoOwner

{

char Name [MAX\_LENGTH\_NAME];

char AutoNum [MAX\_LENGTH\_AUTONUM];

char NumTechPass [MAX\_LENGTH\_NUMTECHPASS];

char Birthdate [MAX\_LENGTH\_BIRTHDATE];

char PhoneNumber [MAX\_LENGTH\_PHONENUMBER];

char OtdGIBDD [MAX\_LENGTH\_OTDGIBDD];

};

void choiceDemand();

void additionStructure ();

void inputStructure ();

void addWord (char \*word\_new, char \*word);

void searchStructure ();

int scanTabulation (char\* StringText);

int shiftText (char \*word, char \*searchValue, int inquiryDemand);

void outputStructure ();

void streamliningStructure ();

void swap(struct AutoOwner \*carStructs1, struct AutoOwner \*carStructs2);

int main ()

{

system ("chcp 1251");

system ("cls");

printf ("Введите номер запроса:\n");

choiceDemand ();

\_getch ();

return 0;

}

void choiceDemand ()

{

printf ("1)Дополнение массива структур;\n");

printf ("2)Поиск структуры с заданым значением;\n");

printf ("3)Вывод структур на экран;\n");

printf ("4)Упорядочение по заданному полю;\n");

printf ("5)Выход из программы;\n");

int inquiry = 0;

scanf ("%d", &inquiry);

switch (inquiry)

{

case 1:

additionStructure ();

break;

case 2:

searchStructure ();

break;

case 3:

outputStructure ();

break;

case 4:

streamliningStructure ();

break;

case 5:

exit (0);

default:

exit (0);

}

}

void additionStructure ()

{

printf ("Введите количество структур\n");

int numberOfStruct = 0;

scanf ("%d", &numberOfStruct);

int i;

for (i = 0; i < numberOfStruct; i++)

{

printf ("Введите структуру № %d \n", i + 1);

inputStructure ();

}

}

void inputStructure ()

{

struct AutoOwner owner;

printf ("Введите имя владельца: ");

scanf ("%s", &owner.Name);

printf ("Введите номер машины: ");

scanf ("%s", &owner.AutoNum);

printf ("Введите номер техпаспорта: ");

scanf ("%s", &owner.NumTechPass);

printf ("Введите дату рождения: ");

scanf ("%s", &owner.Birthdate);

printf ("Введите номер телефона: ");

scanf ("%s", &owner.PhoneNumber);

printf ("Введите отделение ГИБДД: ");

scanf ("%s", &owner.OtdGIBDD);

FILE \*OAiP\_Lab5Var9 = fopen ("OAiP\_Lab5Var9.txt", "a");

if (OAiP\_Lab5Var9 == NULL)

{

fclose (OAiP\_Lab5Var9);

OAiP\_Lab5Var9 = fopen ("OAiP\_Lab5Var9.txt", "w");

}

fprintf (OAiP\_Lab5Var9, "%s\t", owner.Name);

fprintf (OAiP\_Lab5Var9, "%s\t", owner.AutoNum);

fprintf (OAiP\_Lab5Var9, "%s\t", owner.NumTechPass);

fprintf (OAiP\_Lab5Var9, "%s\t", owner.Birthdate);

fprintf (OAiP\_Lab5Var9, "%s\t", owner.PhoneNumber);

fprintf (OAiP\_Lab5Var9, "%s\n\0", owner.OtdGIBDD);

fclose (OAiP\_Lab5Var9);

}

void searchStructure ()

{

printf ("Выберете параметр поиска");

printf ("1) Имя\n");

printf ("2) Номер машины\n");

printf ("3) Номер техпаспорта\n");

printf ("4) Дата рождения\n");

printf ("5) Номер телефона\n");

printf ("6) Отделение ГИБДД\n");

int inquiryDemand = 0;

scanf ("%d", &inquiryDemand);

printf ("Введите значение\n");

char searchValue [MAX\_LENGTH\_OTDGIBDD];

scanf ("%s", &searchValue);

char scanStruct [MAX\_LENGTH];

FILE \*file = fopen ("OAiP\_Lab5Var9.txt", "r");

while (fgets (scanStruct, MAX\_LENGTH, file) != NULL)

{

char scanStruct2 [MAX\_LENGTH];

addWord(scanStruct2, scanStruct);

int i;

if (shiftText (scanStruct2, searchValue, inquiryDemand) == 1)

{

printf ("%s", scanStruct);

}

}

fclose (file);

}

int shiftText (char \*word, char \*searchValue, int inquiryDemand)

{

int i;

for (i = 0; i < (inquiryDemand - 1); i++)

{

word = word + scanTabulation (word) + 1;

}

int length = strlen (searchValue);

for (i = 0; i < length - 1; i++)

{

if (searchValue [i] != word [i]) break;

}

if (i == (length - 1))

{

return 1;

}

return 0;

}

void addWord(char \*word\_new, char \*word)

{

for (int i = 0; i < MAX\_LENGTH; i++)

{

word\_new [i] = '\0';

}

int length = strlen (word);

for (int i = 0; i < length; i++)

{

word\_new [i] = word [i];

}

}

int scanTabulation (char\* StringText)

{

for (int i = 0; StringText [i]; i++)

{

if (StringText [i] == '\t')

return i;

}

return NULL;

}

void outputStructure ()

{

FILE \*OAiP\_Lab5Var9 = fopen ("OAiP\_Lab5Var9.txt", "r");

if (OAiP\_Lab5Var9 == NULL)

{

printf ("Файл не найден/n");

}

else

{

char stringStruct [MAX\_LENGTH];

for (int i = 0; i < MAX\_LENGTH; i++)

{

stringStruct [i] = '\0';

}

while (fgets (stringStruct, MAX\_LENGTH, OAiP\_Lab5Var9) != NULL)

{

printf ("%s", stringStruct);

}

}

fclose(OAiP\_Lab5Var9);

}

void streamliningStructure()

{

FILE \*OAiP\_Lab5Var9 = fopen ("OAiP\_Lab5Var9.txt", "rt");

if (OAiP\_Lab5Var9 == NULL)

{

printf ("Ошибка");

}

else

{

struct AutoOwner carStruct;

int schet = 1;

int nameLength = 0;

while (!feof (OAiP\_Lab5Var9))

{

fscanf(OAiP\_Lab5Var9 , "%s %s %s %s %s %s", carStruct.Name, carStruct.AutoNum, carStruct.NumTechPass,

carStruct.Birthdate, carStruct.PhoneNumber, carStruct.OtdGIBDD);

schet++;

}

rewind(OAiP\_Lab5Var9);

struct AutoOwner \*carStructs = (struct AutoOwner\*)calloc(schet , sizeof(struct AutoOwner));

for (int i = 0; i < schet; i++)

{

fscanf(OAiP\_Lab5Var9, "%s %s %s %s %s %s", carStructs[i].Name, carStructs[i].AutoNum, carStructs[i].NumTechPass,

carStructs[i].Birthdate, carStructs[i].PhoneNumber, carStructs[i].OtdGIBDD);

if (strlen(carStructs[i].Name) > nameLength)

{

nameLength = strlen(carStructs[i].Name);

}

}

for (int k = 0; k < nameLength; k++)

for (int i = 0; i < schet; i++)

for (int j = 0; j < schet - i - 1; j++)

{

if ((int)carStructs[j].Name[k] && (int)carStructs[j + 1].Name[k])

{

if (k == 0)

{

if ((int)carStructs[j].Name[k] > (int)carStructs[j + 1].Name[k]) {

swap(&carStructs[j], &carStructs[j + 1]);

}

}

else

{

if (((int)carStructs[j].Name[k] >(int)carStructs[j + 1].Name[k]) &&((int)carStructs[j].Name[k - 1] == (int)carStructs[j + 1].Name[k -1]))

swap(&carStructs[j], &carStructs[j + 1]);

}

}

}

FILE \*OAiP\_Lab5vAR9\_new = fopen("OAiP\_Lab5vAR9\_new.txt", "w");

for (int i = 0; i < schet; i++)

fprintf(OAiP\_Lab5vAR9\_new, "%s\t%s\t%s\t%s\t%s\t%s\n\0", carStructs[i].Name, carStructs[i].AutoNum, carStructs[i].NumTechPass,

carStructs[i].Birthdate, carStructs[i].PhoneNumber, carStructs[i].OtdGIBDD);

}

}

void swap(struct AutoOwner \*carStructs1, struct AutoOwner \*carStructs2)

{

struct AutoOwner buff = \*carStructs1;

\*carStructs1 = \*carStructs2;

\*carStructs2 = buff;

}

# Тестирование программы

Пример входных и выходных данных, представлены на рисунках 1 и 2

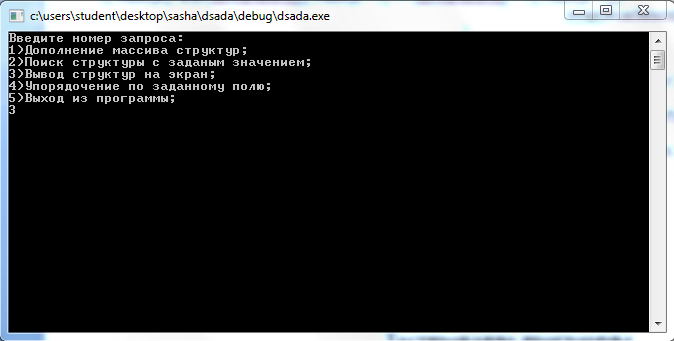


рис.1 Главное меню

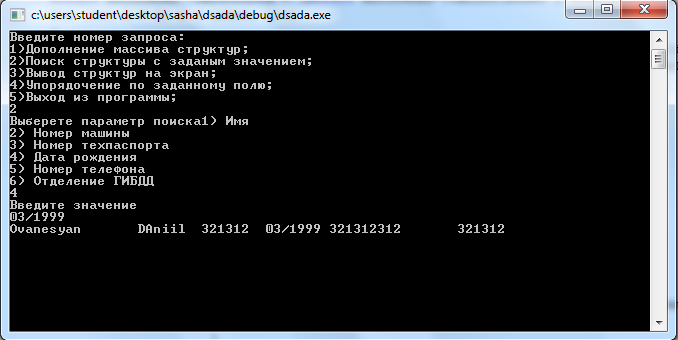
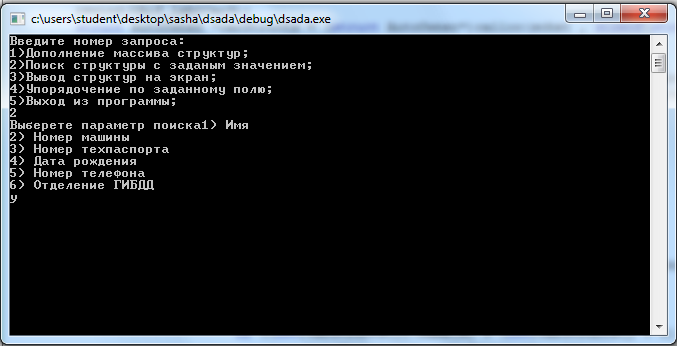


рис.3 Поиск по полю

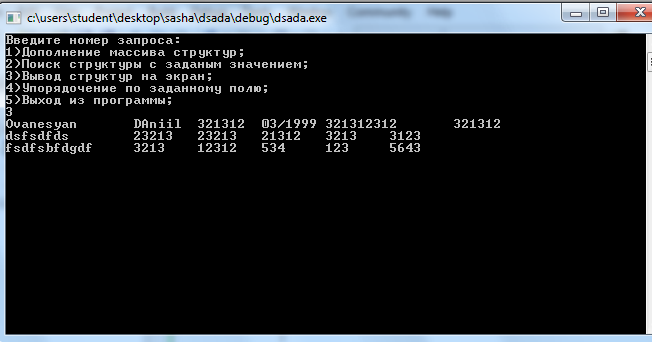


рис.4 Вывод всех структур

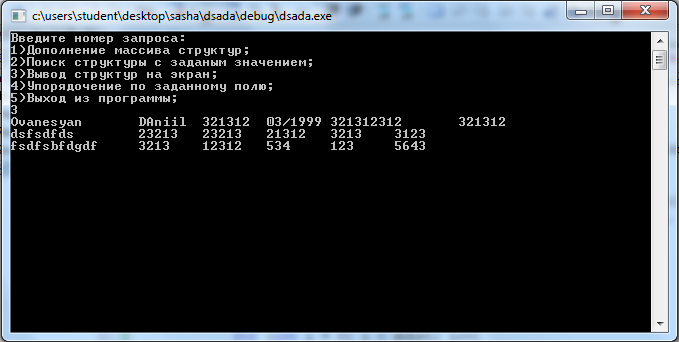


Рис.5 Отсортированная структура

# Вывод

Выполняя данную лабораторной работы, мы ознакомились с понятием структуры, научились правильно использовать и работать с ней.