СОДЕРЖАНИЕ

1. ЗАДАНИЕ 3

2. ОПИСАНИЕ ПРИМЕНЕНИЯ 3

2.1. Запуск программы 3

2.2. Входные данные 3

2.3. Выходные данные 3

2.4. Сообщения программы 4

3. ОПИСАНИЕ ПРОГРАММЫ 4

3.1. Метод решения задачи 4

3.2. Структура программы 5

3.3. Описание функций 5

3.3.1. main – главная функция 5

3.3.2. fio\_nomer – вывод владельцев автомобилей и номера их автомобилей. 7

3.3.3. kolvo\_marki - вывод списка автомобилей каждой марки

и их количества.  8

ЛИТЕРАТУРА 11

Приложение 1. Текст программы 12

Приложение 2. Пример входного файла avto.txt 17

Приложение 3. Результаты тестирования программы 17

**1. ЗАДАНИЕ**

Дан файл, содержащий сведения о владельцах автомобилей. Структура записи файла: фамилия и инициалы владельца, марка и номер автомобиля.

Написать программу, которая печатает

- фамилии владельцев и номера автомобилей данной марки;

- количество автомобилей каждой марки.

**2. ОПИСАНИЕ ПРИМЕНЕНИЯ**

**2.1 Запуск программы**

Запуск программы производится в среде программирования CodeBlocks

**2.2 Входные данные**

Входные данные находятся в файле “avto.txt”. Число строк в файле - произвольное. Каждая строка содержит фамилию с инициалами (20 символов), марку машины (15 символов), номер машины (6 символов). Пример строки файла:

Иванов И. И., LADA А132БР.

| 20 | 15 | 7 |

Пример входного файла приведен в приложении 2.

По запросу программы с клавиатуры необходимо ввести номер пункта меню (см. раздел 2.3).

**2.3. Выходные данные**

Программа выводит на экран меню:

===========================================================

**Выберите номер пункта меню:**

**1 - вывод списка владельцев автомобилей и их номеров**

**2 - вывод марок автомобилей и их количество**

**3 - выход**

При выборе пункта 1 на экран выводится результат в виде:

**№ Фамилия И.О. № Машины**

**--------------------------------------------------**

**1 Иванов И. И. А132БР**

**2 Гараев Л. З. П735КЗ**

**3 Харламов В. Н. К406НН**

**4 Воробушкин Н. К. К947ЛР**

**5 Вафин Н. К. О384ШГ**

**6 Анисин А. К. Д182ГЛ**

**7 Зарипов Д. Р. Х487ВВ**

**8 Морозов А. К. Д787РТ**

**9 Харитонов Д. П. Т503КА**

**Вывод завершен**

При отсутствии файла выводится сообщение:

Файл не может быть открыт!

Примеры выходных данных для остальных пунктов меню см. в приложении 3.

**2.4. Сообщения программы**

Ниже приводится перечень возможных сообщений программы:

**1.Файл не может быть открыт!**

**2.Выберите номер пункта меню:**

**3.Нужно вводить номер пункта от 1 до 4**

**4.Для продолжения нажмите любую клавишу**

**5.Файл пуст!**

**6.№ Фамилия И.О. № Машины**

**7.Вывод завершен**

**8.№ Марка Кол-во**

**3. ОПИСАНИЕ ПРОГРАММЫ**

**3.1. Метод решения задачи**

Задачу можно разбить на две отдельные подзадачи:

1. Вывод Фамилии И. О. владельцев автомобилей и номера этих автомобилей.

2. Подсчет и вывод количества автомобилей каждой марки.

Каждая подзадача решается методом линейного поиска (последовательного просмотра элементов таблицы).

Чтобы пользователь мог выбирать, какие подзадачи решать и в каком порядке, программа выводит на экран меню (см. раздел 2.3).

**3.2. Структура программы**

kolvo\_marki

fio\_nomer

main

Рис.1. Функциональная структура программы

Программа состоит из трех функций: главной функции main и двухподпрограмм.

fio\_nomer - выводит Фамилию И. О. владельца автомобиля и номер этого автомобиля.

kolvo\_marki - выводит количество автомобилей каждой марки.

**3.3. Описание функций**

**3.3.1. main – главная функция**

Заголовок функции:

int main ()

Значение функции:

0 – в случае успешного завершения;

1 – если входной файл не найден.

Рабочие данные:

f – объект класса ifstream, с помощью которого получают доступ к указанному файлу;

n – номер пункта меню.

n

чтение номера пункта меню n

вывод меню

f=NULL

ifstream f(”avto.txt”)

да нет

вывод “Файл не может быть открыт!”

fio\_nomer

'1'

kolvo\_marki

'2'

n!='3'

да

нет

закрытие файла

Рис.1 Блок-схема функции main()

**3.3.2. fio\_nomer – вывод владельцев автомобилей и номера их автомобилей.**

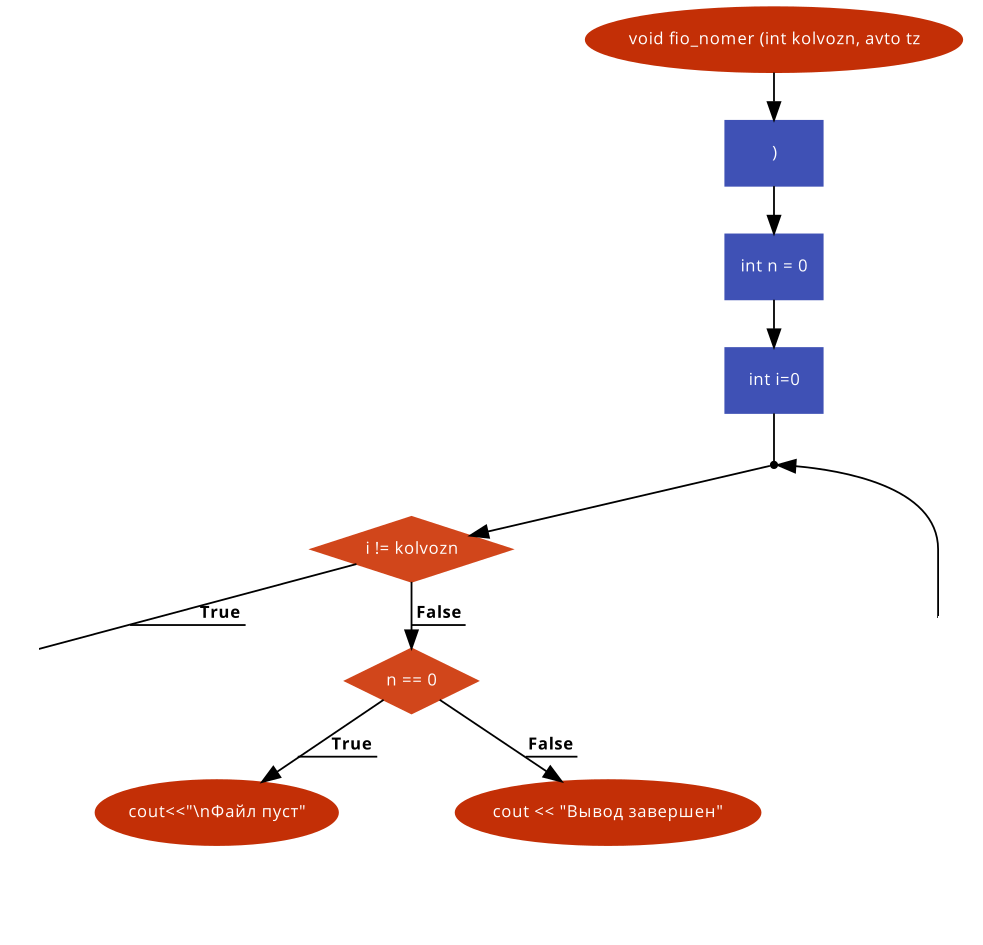
Заголовок функции: Структура записи файла:

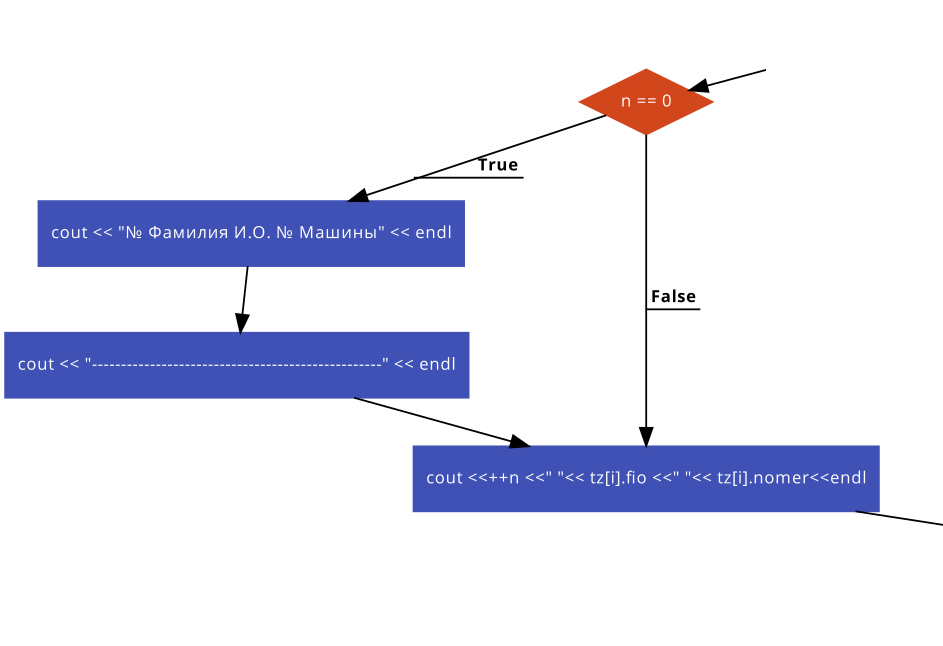
void fio\_nomer (int kolvozn, avto tz[])

fio - фамилия и инициалы владельца nomer - номер автомобиля

Входные данные: marka - марка автомобиля

f - объект класса ifstream, с помощью которого получают доступ к указанному файлу





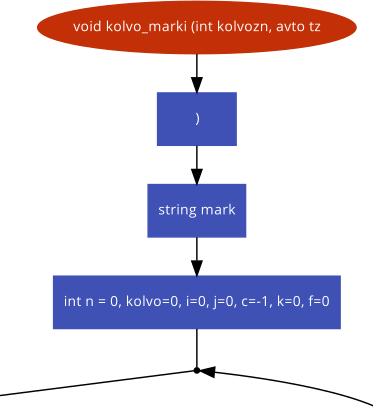
i++

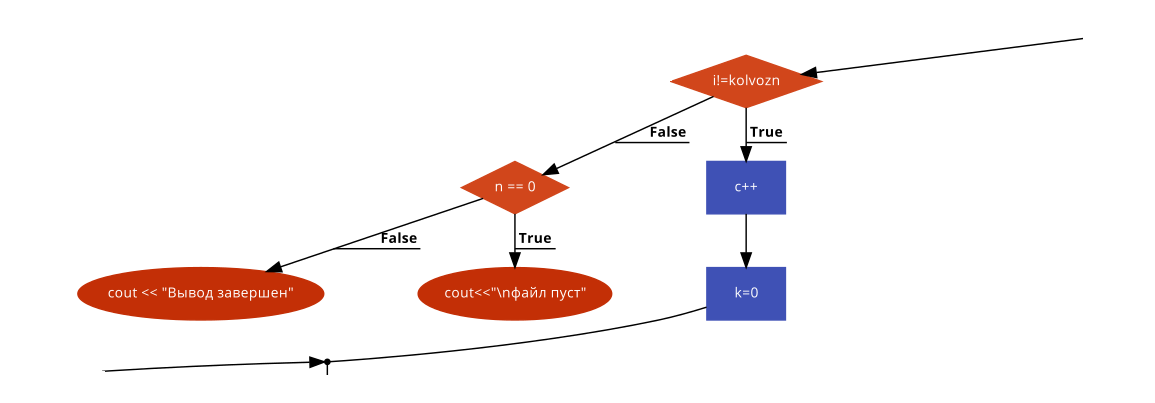
Рис.2. Схема функции вывода списка владельцев автомобилей и номеров их автомобилей.

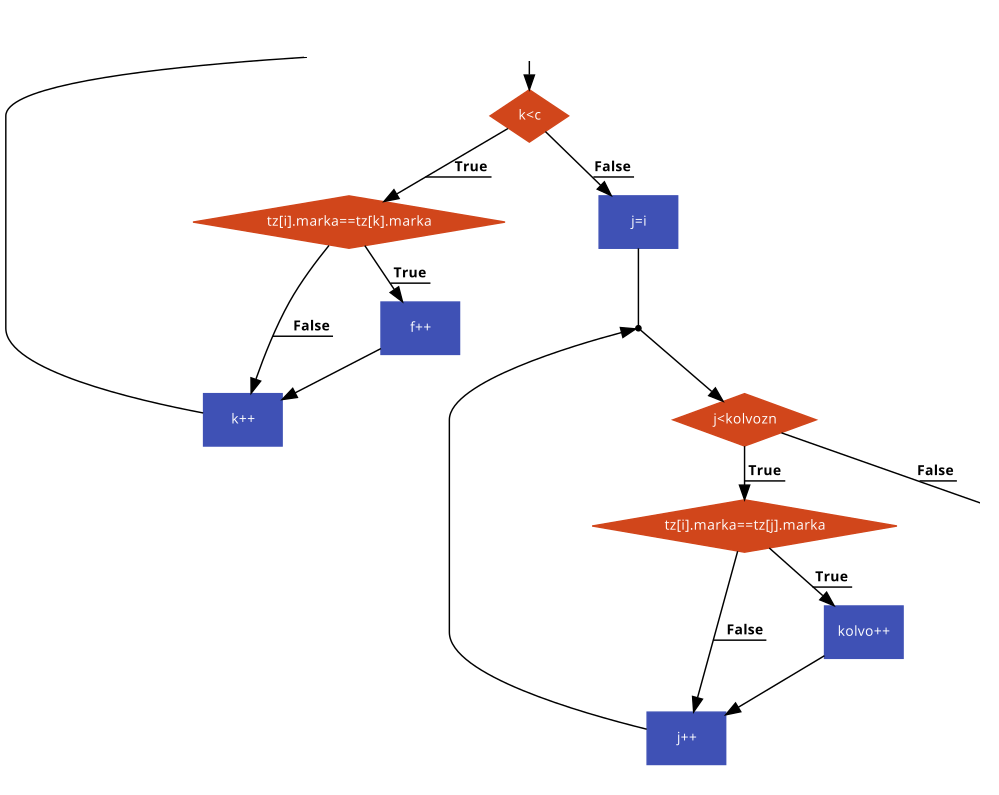
**3.3.3. kolvo\_marki - вывод списка автомобилей каждой марки и их количества.**

Заголовок функции:

void kolvo\_marki (int kolvozn, avto tz[])







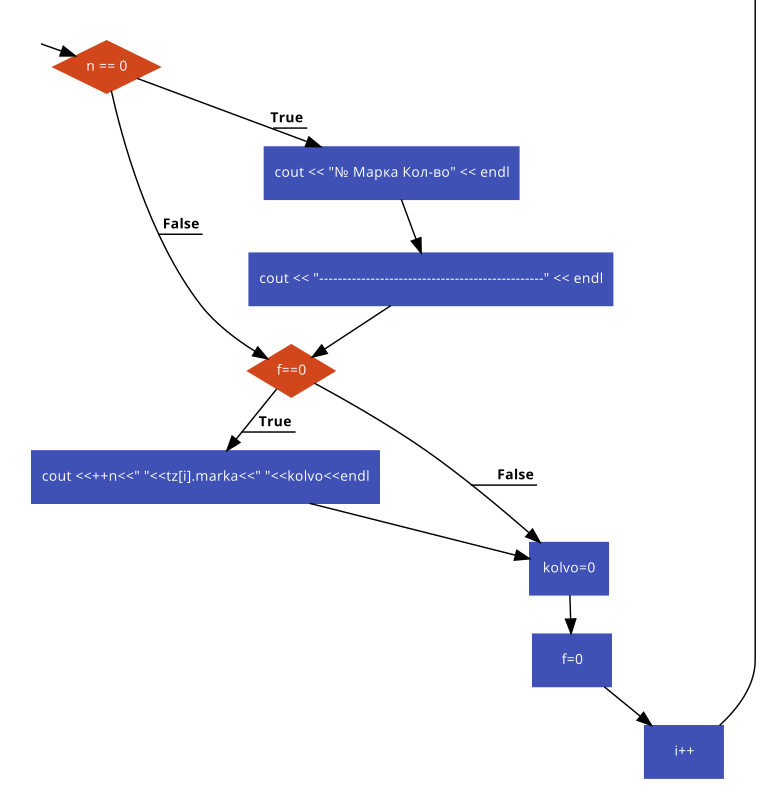


Рис.3. Блок-схема функции вывода списка автомобилей каждой марки и их количества.

**Литература.**

1. Задания к курс. работе.

2. Пример отчета по курсовой работе.

3. Пример титульного листа.

4. Пример кода программы.

**Приложение 1. Текст программы**

#include <fstream>

#include <string>

#include <iostream>

#include <sstream>

#include <windows.h>

using namespace std;

struct avto

{

string fio;

string marka;

string nomer;

};

void fio\_nomer (int kolvozn, avto tz[]);

void kolvo\_marki (int kolvozn, avto tz[]);

int main()

{

ifstream f; /\* ссылка на входной файл \*/

char n; /\* номер пункта меню \*/

int i;

char sl;

string sl1;

SetConsoleCP(1251);// установка кодовой страницы win-cp 1251 в поток ввода

SetConsoleOutputCP(1251); // установка кодовой страницы win-cp 1251 в поток вывода

f.open("avto.txt");

if (!f.is\_open())

{

cout << "Файл не может быть открыт!\n";

cin.get();

return 1;

}

else if (!f.eof())

{

avto tz[88];

for (i = 0; f.good();i++)

{

if ((sl=f.get())=='\n'&&i!=0)

getline(f, tz[i].fio,',');

else

{

tz[i].fio += sl;

getline(f, sl1, ',');

tz[i].fio += sl1;

}

f >> tz[i].marka;

if ((sl = f.get()) == ' ')

getline(f, tz[i].nomer,'.');

cout << tz[i].fio<<" "<< tz[i].marka <<" " << tz[i].nomer<<" "<<endl;

}

int kolvozn = i - 1;

do

{

cout << "========================================================"<<endl;

cout << "Выберите номер пункта меню:"<<endl;

cout << "1 - вывод списка владельцев автомобилей и их номеров" << endl;

cout << "2 - вывод марок автомобилей и их количество"<<endl;

cout << "3 - выход"<<endl;

cout << "------------------------------------------------------"<<endl;

n=cin.get();

cin.sync();

switch (n)

{

case '1': fio\_nomer(kolvozn,tz); break;

case '2': kolvo\_marki(kolvozn,tz); break;

case '3': break;

default: cout << "\nНужно вводить номер пункта от 1 до 4";

}

if (n != '3')

{

cout << "\nДля продолжения нажмите любую клавишу";

cin.get();

}

} while (n != '3');

f.close();

}

else

cout << "Файл пуст!\n";

//cin.get();

return 0;

}

void fio\_nomer (int kolvozn, avto tz[])

{

int n = 0;

int i=0;

while (i != kolvozn)

{

if (n == 0)

{

cout << "№ Фамилия И.О. № Машины" << endl;

cout << "--------------------------------------------------" << endl;

}

cout <<++n <<" "<< tz[i].fio <<" "<< tz[i].nomer<<endl;

i++;

}

if (n == 0) cout<<"\nФайл пуст";

else cout << "Вывод завершен";

}

void kolvo\_marki (int kolvozn, avto tz[])

{

string mark;

int n = 0, kolvo=0, i=0, j=0, c=-1, k=0, f=0;

while (i!=kolvozn)

{

c++;

for (k=0;k<c;k++)

if(tz[i].marka==tz[k].marka)

f++;

for (j=i; j<kolvozn; j++)

if (tz[i].marka==tz[j].marka)

kolvo++;

if (n == 0)

{

cout << "№ Марка Кол-во" << endl;

cout << "------------------------------------------------" << endl;

}

if (f==0)

cout <<++n<<" "<<tz[i].marka<<" "<<kolvo<<endl;

kolvo=0;

f=0;

i++;

}

if (n == 0) cout<<"\nфайл пуст";

else cout << "Вывод завершен";

}

**Приложение 2. Пример входного файла avto.txt**

Иванов И. И., LADA А132БР.

Гараев Л. З., Chevrolet П735КЗ.

Харламов В. Н., Mercedes-Benz К406НН.

Воробушкин Н. К., Datsun К947ЛР.

Вафин Н. К., LADA О384ШГ.

Анисин А. К., Ford Д182ГЛ.

Зарипов Д. Р., BMW Х487ВВ.

Морозов А. К., Mercedes-Benz Д787РТ.

Харитонов Д. П., Ford Т503КА.

**Приложение 3. Результаты тестирования программы**

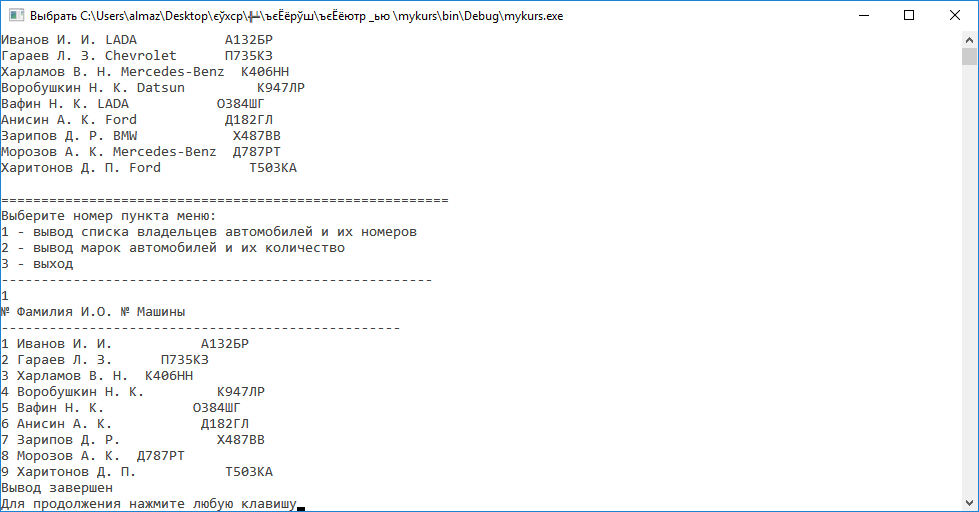
**Тест 1.** Входного файла нет в текущем каталоге.

Результат:

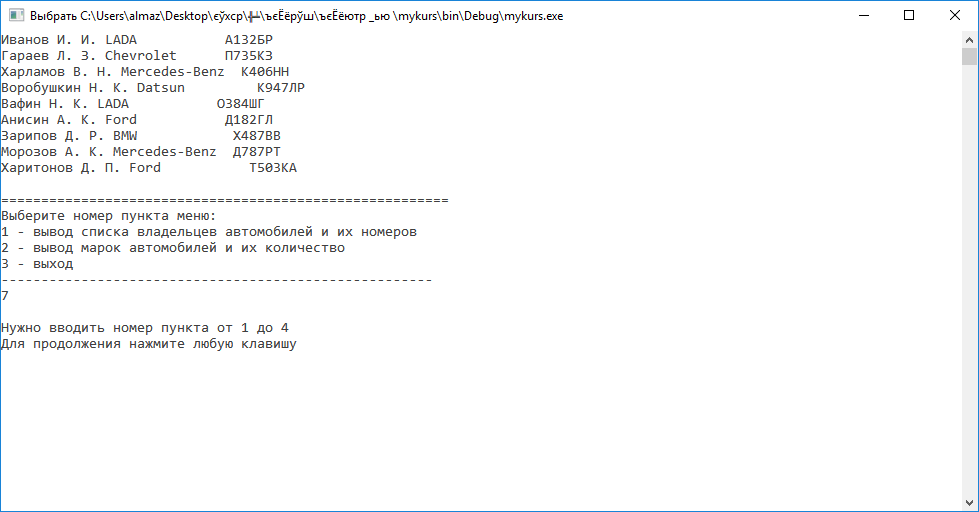


В следующих тестах используется файл из приложения 2.

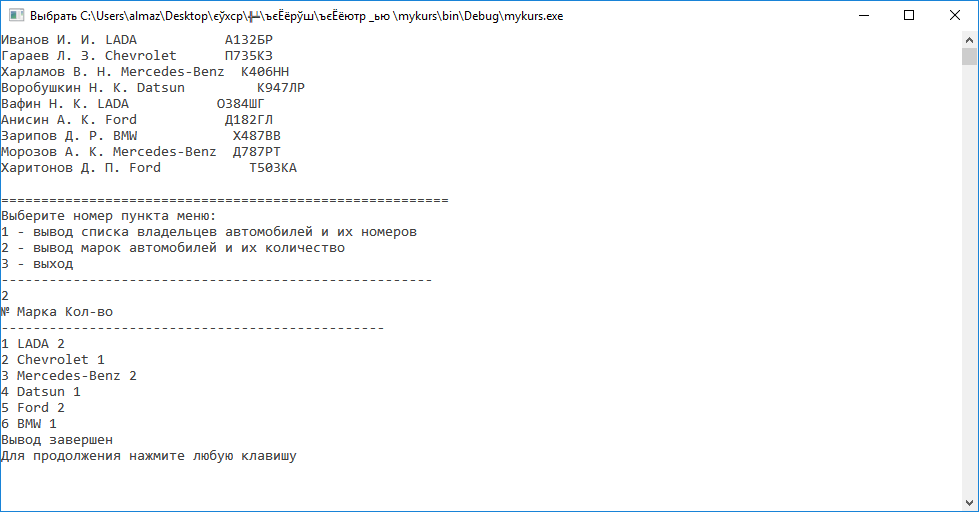
**Тест 2.**



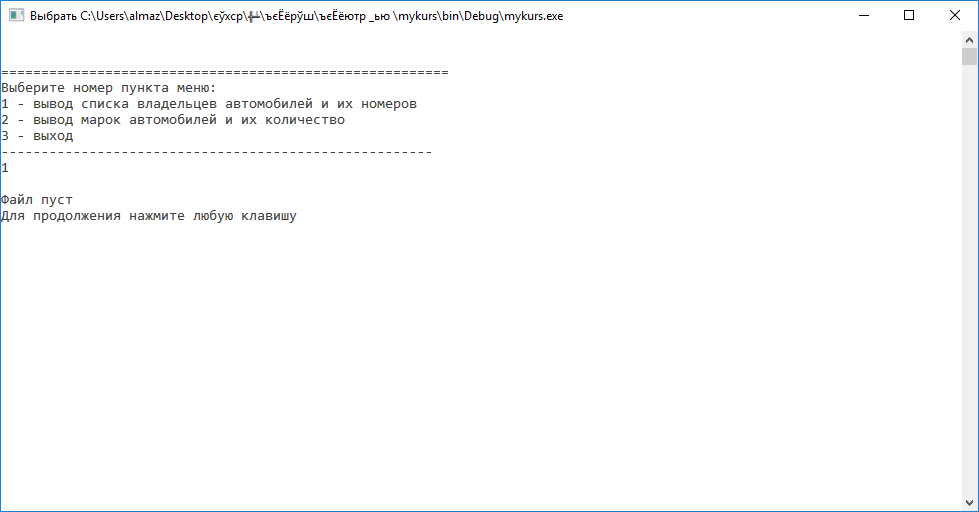
**Тест 3.**

****

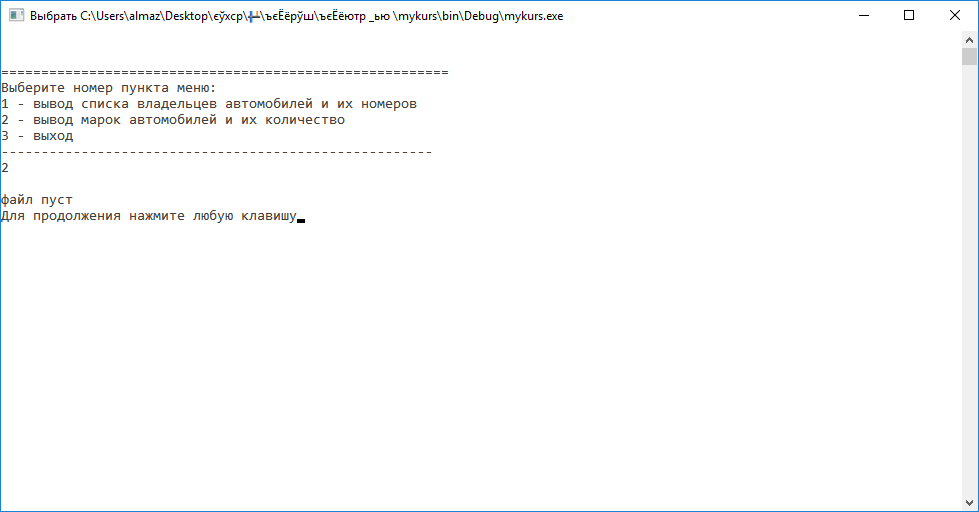
**Тест 4.**

****

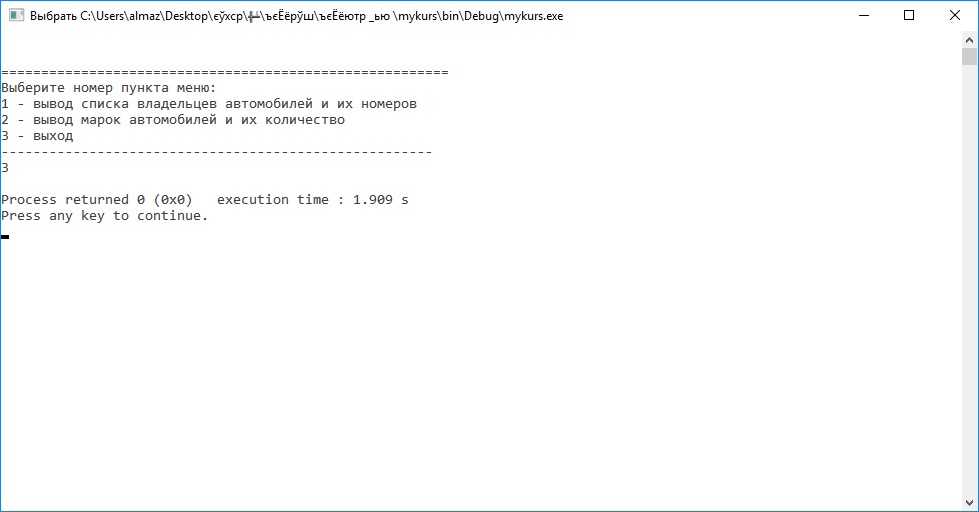
**Тест 5.**

****

**Тест 6.**

****

**Тест 7.**

****