Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе № 8**

Дисциплина: «Информатика»

Тема: Структуры данных

Вариант 21

Выполнил:

Студент группы РИС-20-1б

Ознобихин Елисей Андреевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

Работа со структурами, ввод-вывод информации и хранение её в полях элементов структуры

**Постановка задачи**

Структура "Автомобиль":

- марка;

- серийный номер;

- регистрационный номер;

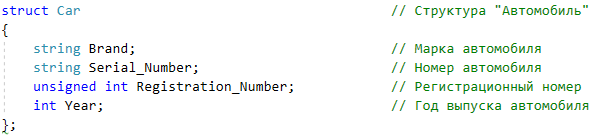
- год выпуска.

Удалить 3 элемента из начала файла, добавить элемент поле элемента с указанным регистрационным номером.

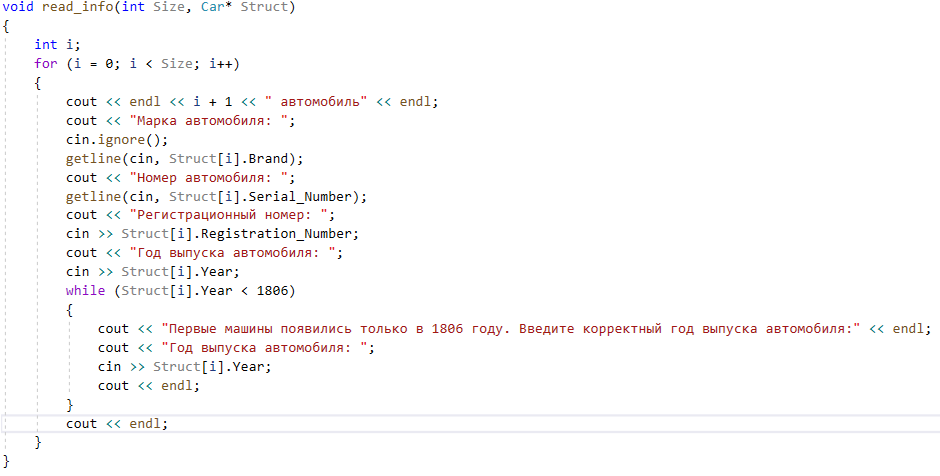
**Анализ задачи**

**1.** Для решения задачи необходимо:

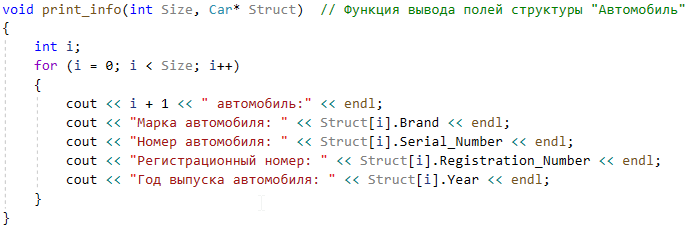
**1.1.** Зарезервировать память под структуру с заранее известным количеством элементов и инициализировать её методом ручного ввода данных.



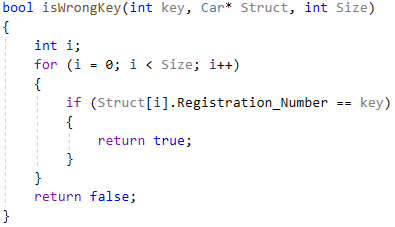
**1.2.** Разработать функцию поэлементного ввода данных в структуру с проверкой на возможные ошибки.



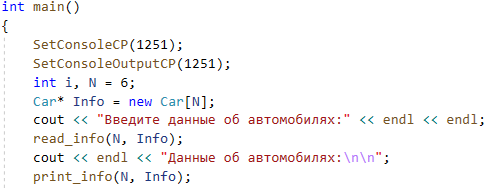
**1.3.** Разработать функцию поэлементного вывода данных структуры на консоль.



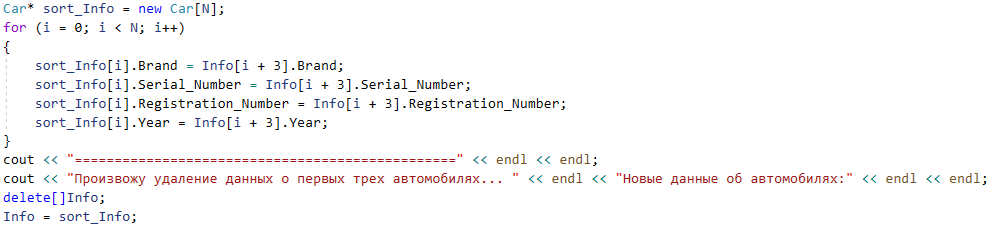
**1.4.** Разработать функцию для проверки правильно введенного ключа поиска данных в структуре.



**1.5.** Реализовать применение этих функций в программе.



**1.6.** Реализовать добавление и удаление элементов структуры.



**2.** В ходе работы были использованы типы данных:

**2.1.** Структура Car для хранения данных и работы с ними.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_9BsmOoinvl.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_5caLebrAWq.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_RikTIJBVPT.png

**2.2.** Для функции print\_info используются следующие аргументы:

1. Тип int: размерность структуры.

2. Указатель на Car: структура данных.

Сама функция имеет тип void, так как нет необходимости возвращать значение для этой функции.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_L1RJqtJnLS.png

**2.3.** Для функции read\_info используются следующие аргументы:

1. Тип int: размерность структуры.

2. Указатель на Car: структура данных.

Сама функция имеет тип void, так как нет необходимости возвращать значение для этой функции.

**C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_PniMDVYMrl.png**

**2.4.** Для функции isWrongKey используются следующие аргументы:

1. Тип int: ключ поиска данных в структуре.

2. Указатель на Car: структура данных.

3. Тип int: размерность структуры.

Сама функция имеет тип bool для передачи информации о том, совпал ли ключ поиска с одним из полей элемента структуры.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_6Dx6wtuq5D.png

**3.** Для решения задачи данные были представлены в следующем виде:

**3.1.** Для осуществления работы данные были записаны ручным вводом с консоли в виде структуры.

**4.** Для операций ввода и вывода использовались следующие операторы и функции:

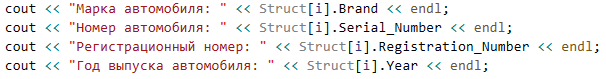
**4.1.** Ввод данных в структуру реализован с помощью оператора ввода cin и функции getline.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_pcwfe6BpUM.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_gWlCYGP5a0.png

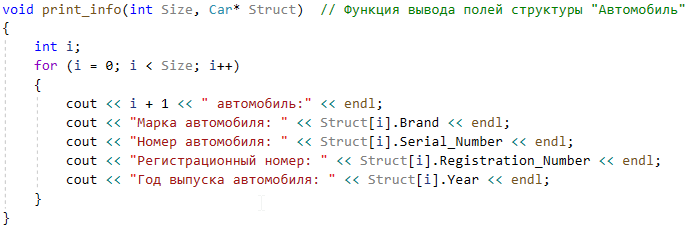
**4.2.** Вывод данных на консоль реализован с помощью оператора cout.

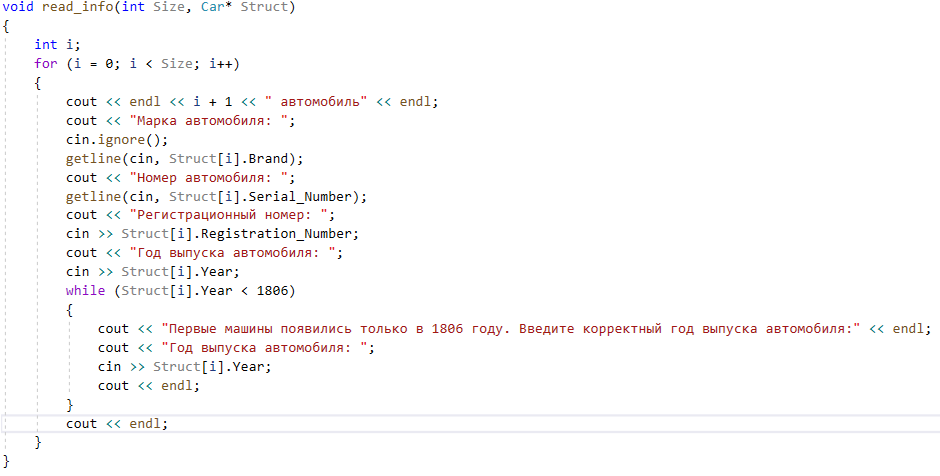
C:\Users\Асус\Documents\ShareX\Screenshots\2021-02\devenv_3Gd5Q3bsYw.png



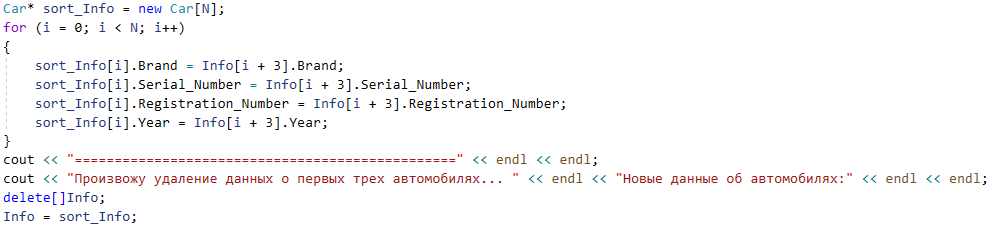
**5.** Поставленные задачи решены следующими действиями:

**5.1.** Ввод и вывод данных из структуры реализованы с помощью войдовых функций print\_info и read\_info.

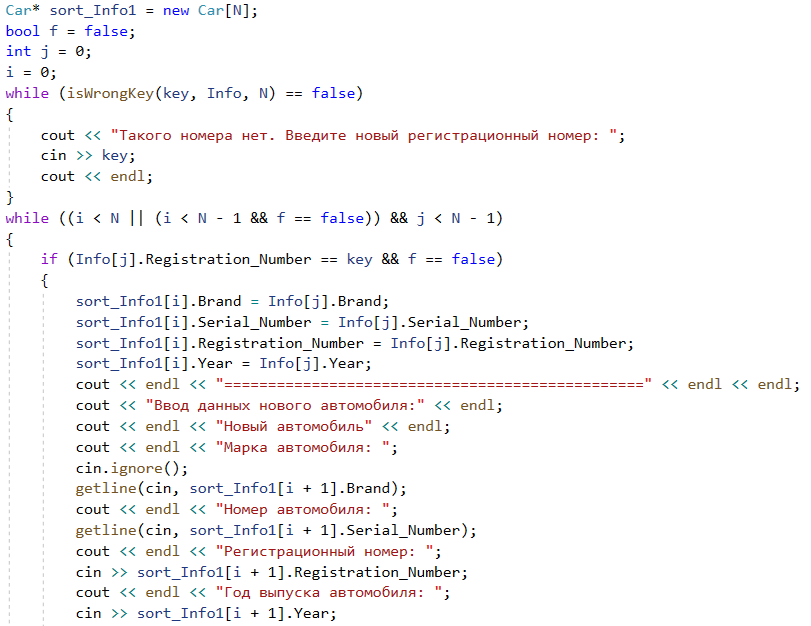


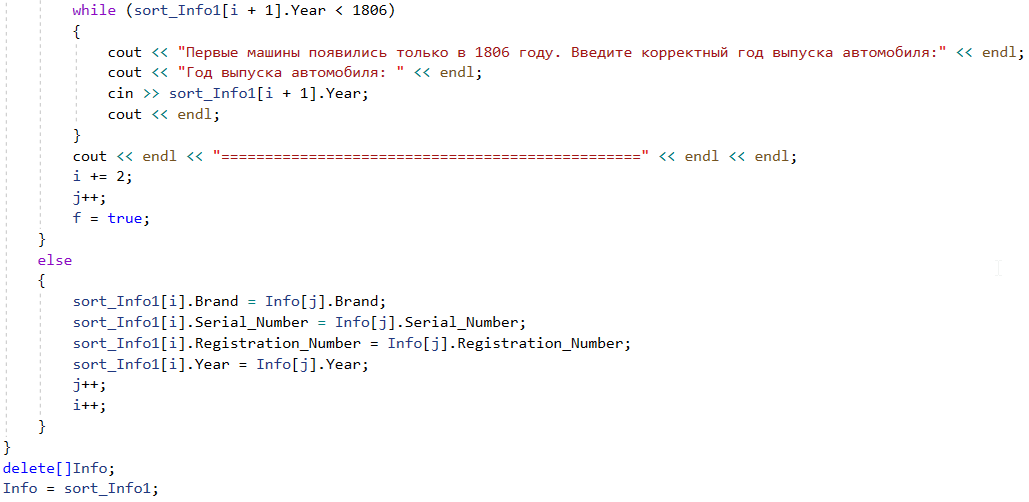


**5.2.** Иллюзия удаления первых трёх элементов из структуры реализована с помощью основной структуры и дополнительной: элементы основной структуры записываются в дополнительную структуру, начиная с третьего элемента из основной структуры в качестве первого элемента дополнительной структуры. После завершения переноса необходимых элементов из основной структуры в дополнительную освобождаем память под основную структуру и приравниваем указатели основной и дополнительной структуры.

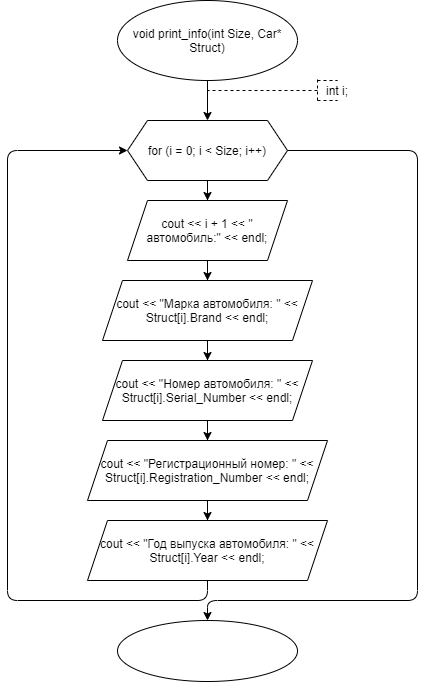


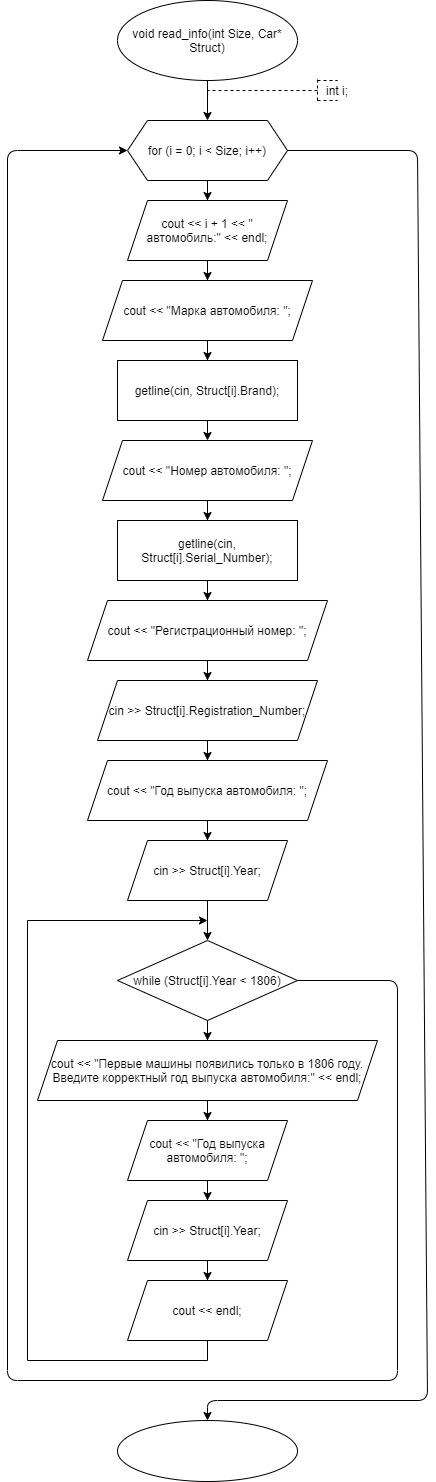
**5.3.** Иллюзия добавления элемента в структуру реализована с помощью основной структуры и дополнительной: элементы основной структуры записываются в дополнительную структуру, начиная с первого элемента из основной структуры в качестве первого элемента дополнительной структуры. По нахождению ключа поиска данных производится ручной ввод нового элемента структуры и переноса элементов. После завершения переноса необходимых элементов из основной структуры в дополнительную освобождаем память под основную структуру и приравниваем указатели основной и дополнительной структуры.

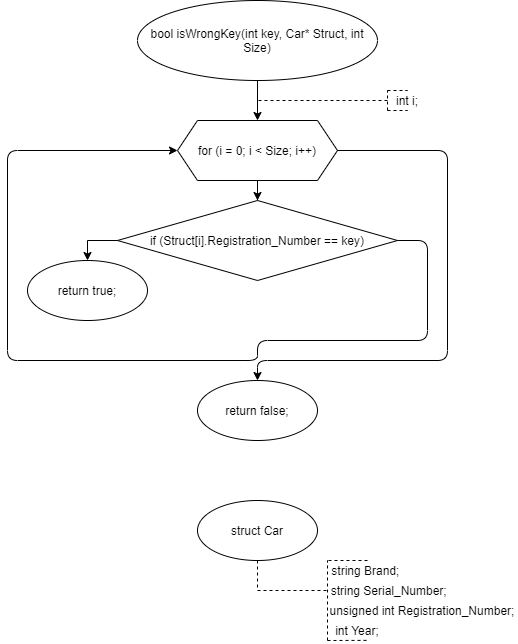


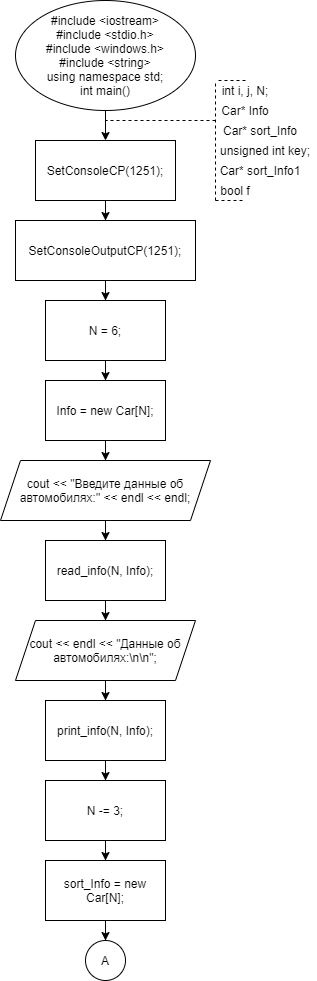


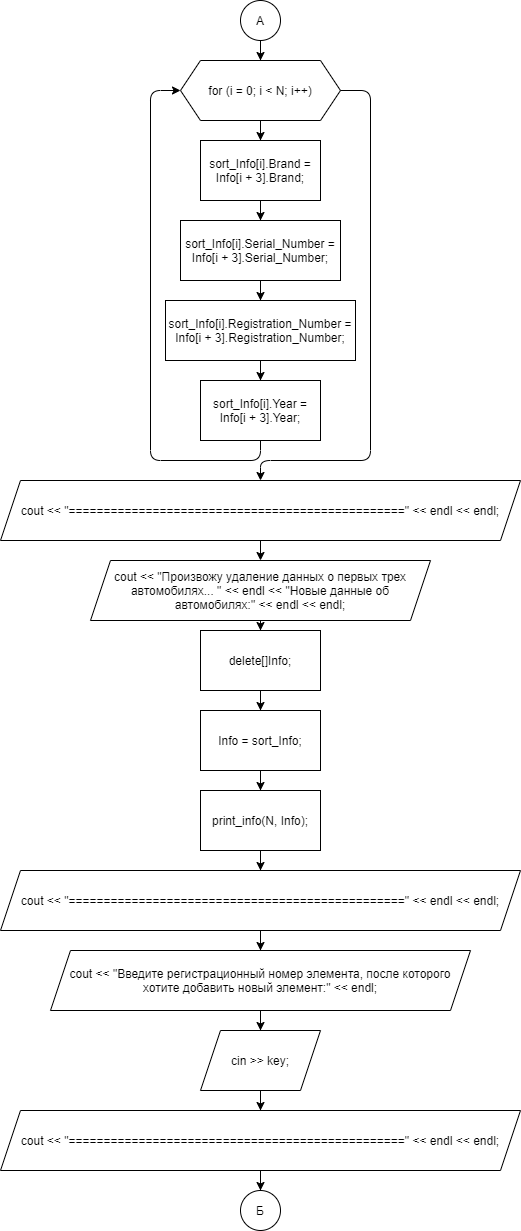
**Блок-схема программы**

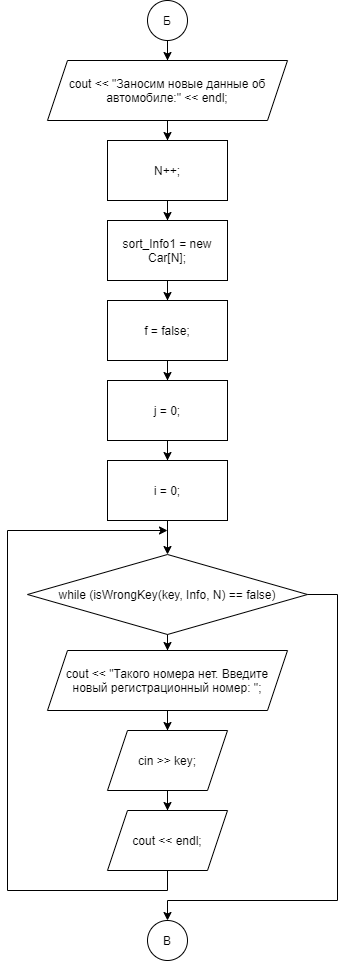
****

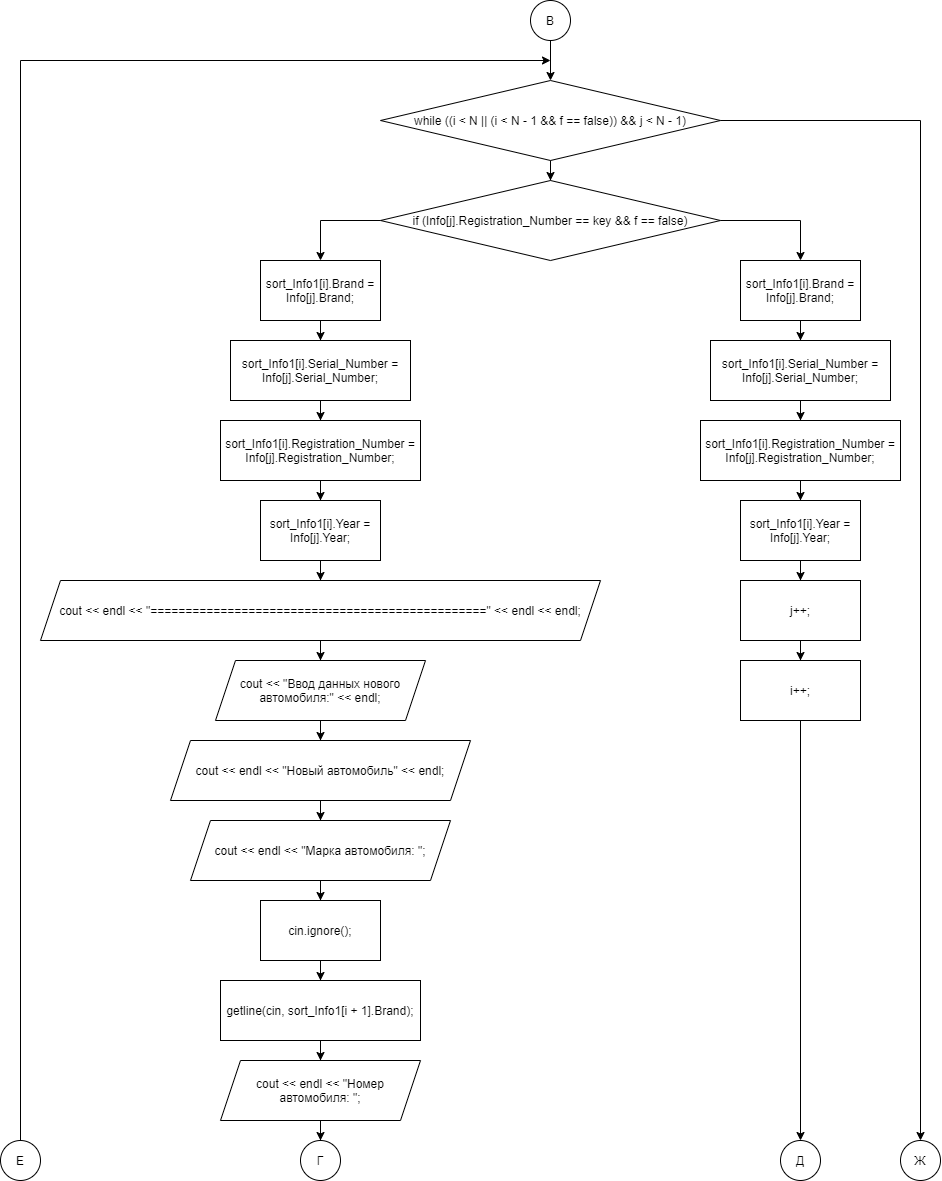
****

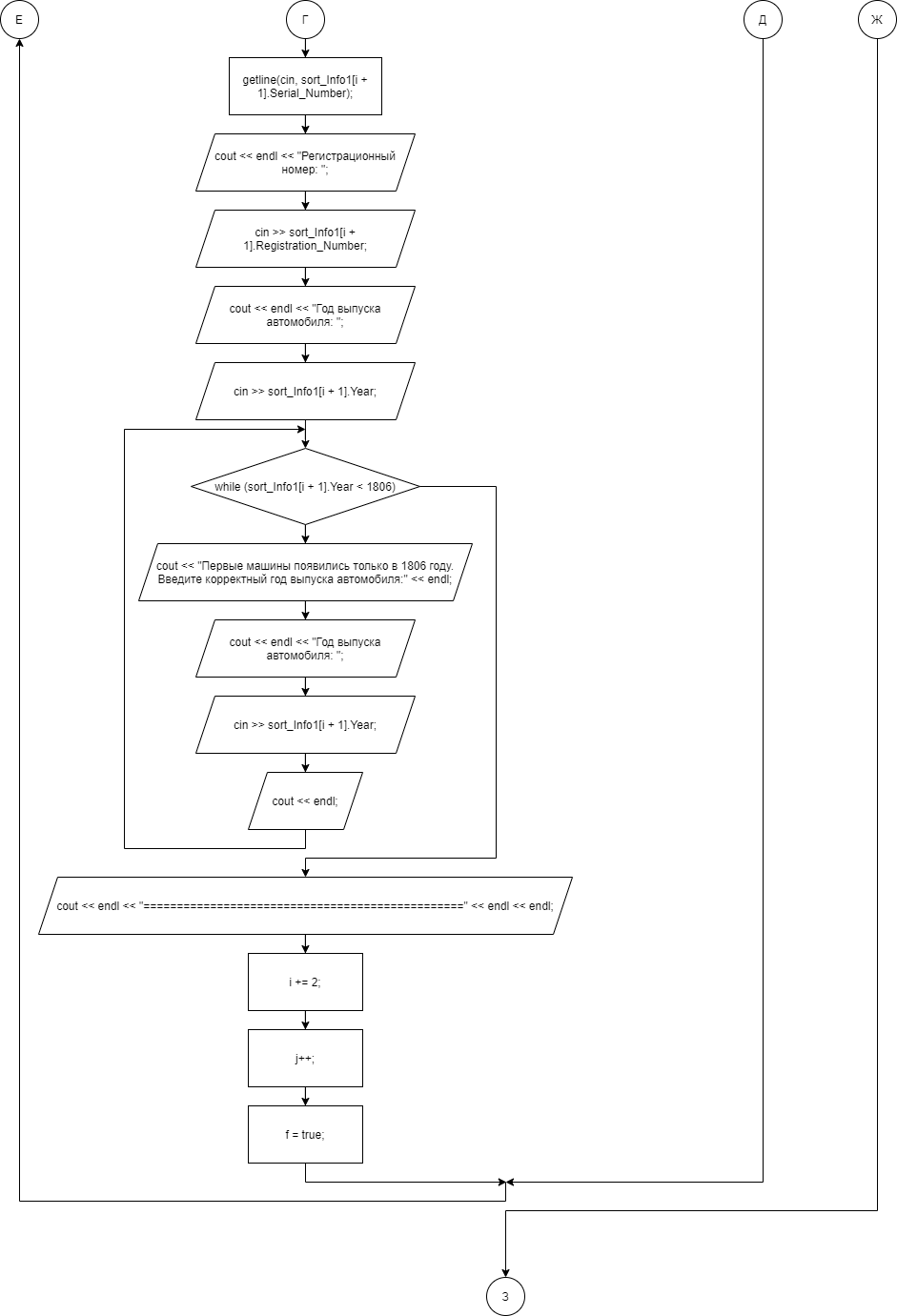
****

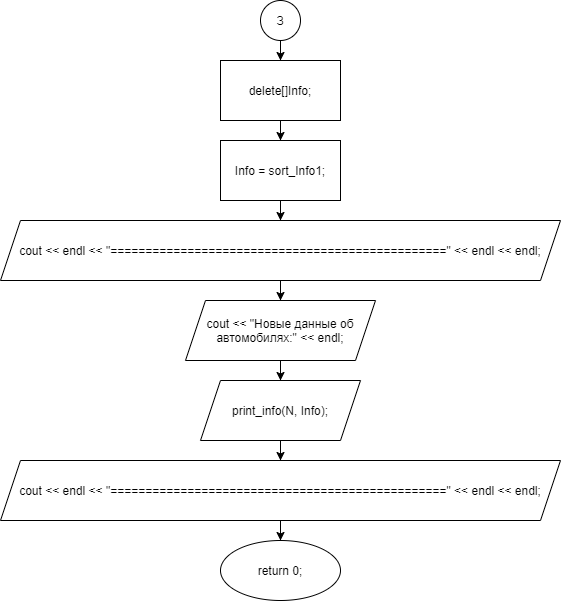
****

****

****

****

****

****

**Реализация задачи на языке С++**

#include <iostream>

#include <stdio.h>

#include <windows.h>

#include <string>

using namespace std;

struct Car // Структура "Автомобиль"

{

string Brand; // Марка автомобиля

string Serial\_Number; // Номер автомобиля

unsigned int Registration\_Number; // Регистрационный номер

int Year; // Год выпуска автомобиля

};

void print\_info(int Size, Car\* Struct)

{

int i;

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

{

cout << i + 1 << " автомобиль:" << endl;

cout << "Марка автомобиля: " << Struct[i].Brand << endl;

cout << "Номер автомобиля: " << Struct[i].Serial\_Number << endl;

cout << "Регистрационный номер: " << Struct[i].Registration\_Number << endl;

cout << "Год выпуска автомобиля: " << Struct[i].Year << endl;

}

}

void read\_info(int Size, Car\* Struct)

{

int i;

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

{

cout << endl << i + 1 << " автомобиль" << endl;

cout << "Марка автомобиля: ";

cin.ignore();

getline(cin, Struct[i].Brand);

cout << "Номер автомобиля: ";

getline(cin, Struct[i].Serial\_Number);

cout << "Регистрационный номер: ";

cin >> Struct[i].Registration\_Number;

cout << "Год выпуска автомобиля: ";

cin >> Struct[i].Year;

while (Struct[i].Year < 1806)

{

cout << "Первые машины появились только в 1806 году. Введите корректный год выпуска автомобиля:" << endl;;

cout << "Год выпуска автомобиля: ";

cin >> Struct[i].Year;

cout << endl;

}

cout << endl;

}

}

bool isWrongKey(int key, Car\* Struct, int Size)

{

int i;

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

{

if (Struct[i].Registration\_Number == key)

{

return true;

}

}

return false;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int i, N = 6;

Car\* Info = new Car[N];

cout << "Введите данные об автомобилях:" << endl << endl;

read\_info(N, Info);

N -= 3;

Car\* sort\_Info = new Car[N];

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

{

sort\_Info[i].Brand = Info[i + 3].Brand;

sort\_Info[i].Serial\_Number = Info[i + 3].Serial\_Number;

sort\_Info[i].Registration\_Number = Info[i + 3].Registration\_Number;

sort\_Info[i].Year = Info[i + 3].Year;

}

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

cout << "Произвожу удаление данных о первых трех автомобилях... " << endl << "Новые данные об автомобилях:" << endl << endl;;

delete[]Info;

Info = sort\_Info;

print\_info(N, Info);

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

cout << "Введите регистрационный номер элемента, после которого хотите добавить новый элемент:" << endl;

unsigned int key;

cin >> key;

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

cout << "Заносим новые данные об автомобиле:" << endl;

N++;

Car\* sort\_Info1 = new Car[N];

bool f = false;

int j = 0;

i = 0;

while (isWrongKey(key, Info, N) == false)

{

cout << "Такого номера нет. Введите новый регистрационный номер: ";

cin >> key;

cout << endl;

}

while ((i < N || (i < N - 1 && f == false)) && j < N - 1)

{

if (Info[j].Registration\_Number == key && f == false)

{

sort\_Info1[i].Brand = Info[j].Brand;

sort\_Info1[i].Serial\_Number = Info[j].Serial\_Number;

sort\_Info1[i].Registration\_Number = Info[j].Registration\_Number;

sort\_Info1[i].Year = Info[j].Year;

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

cout << "Ввод данных нового автомобиля:" << endl;

cout << endl << "Новый автомобиль" << endl;

cout << endl << "Марка автомобиля: ";

cin.ignore();

getline(cin, sort\_Info1[i + 1].Brand);

cout << endl << "Номер автомобиля: ";

getline(cin, sort\_Info1[i + 1].Serial\_Number);

cout << endl << "Регистрационный номер: ";

cin >> sort\_Info1[i + 1].Registration\_Number;

cout << endl << "Год выпуска автомобиля: ";

cin >> sort\_Info1[i + 1].Year;

while (sort\_Info1[i + 1].Year < 1806)

{

cout << "Первые машины появились только в 1806 году. Введите корректный год выпуска автомобиля:" << endl;;

cout << "Год выпуска автомобиля: " << endl;

cin >> sort\_Info1[i + 1].Year;

cout << endl;

}

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

i += 2;

j++;

f = true;

}

else

{

sort\_Info1[i].Brand = Info[j].Brand;

sort\_Info1[i].Serial\_Number = Info[j].Serial\_Number;

sort\_Info1[i].Registration\_Number = Info[j].Registration\_Number;

sort\_Info1[i].Year = Info[j].Year;

j++;

i++;

}

}

delete[]Info;

Info = sort\_Info1;

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

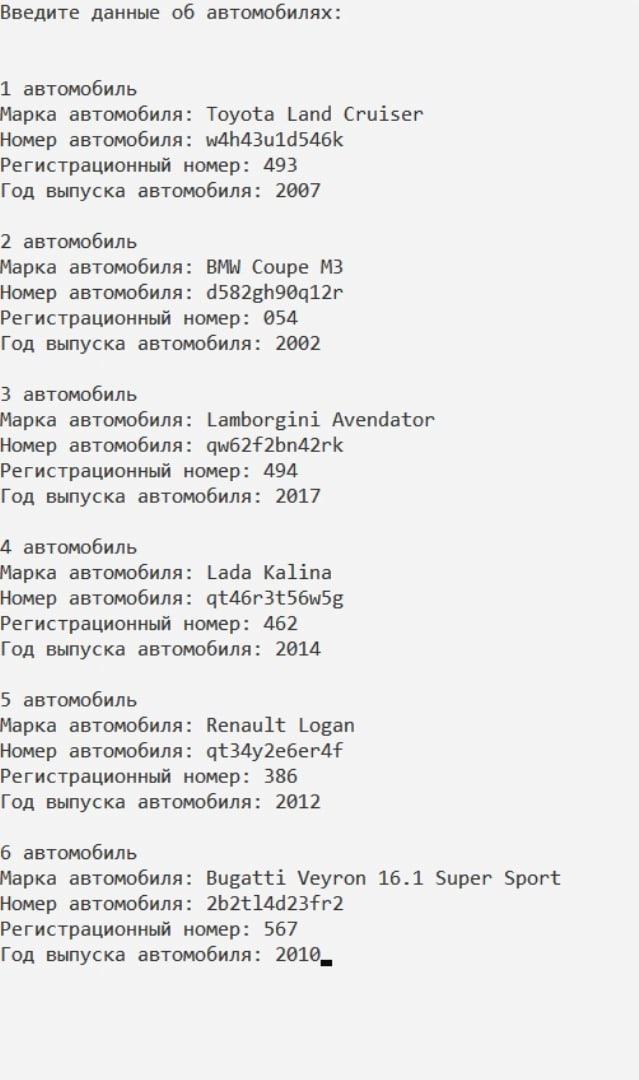
cout << "Новые данные об автомобилях:" << endl;

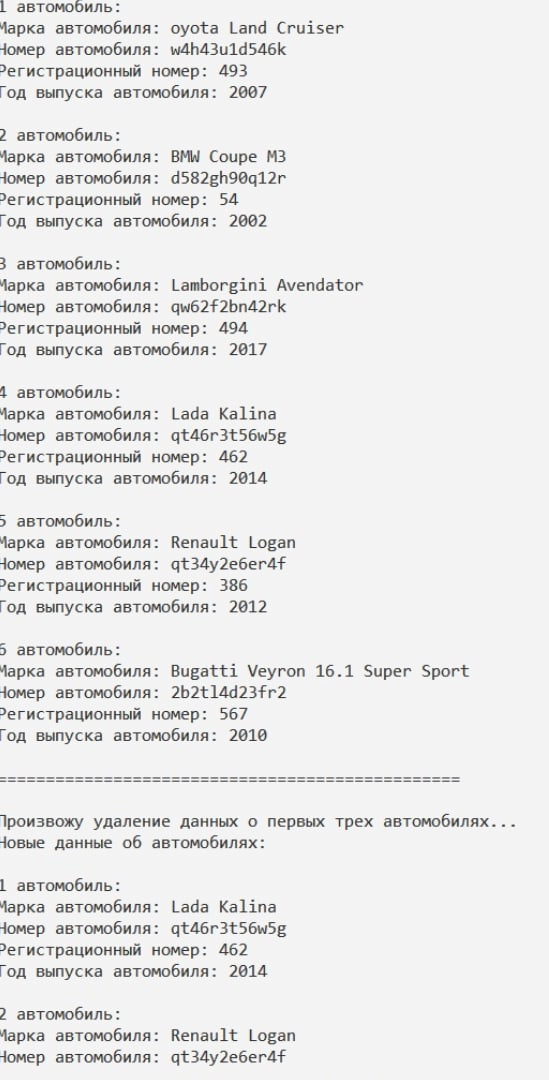
print\_info(N, Info);

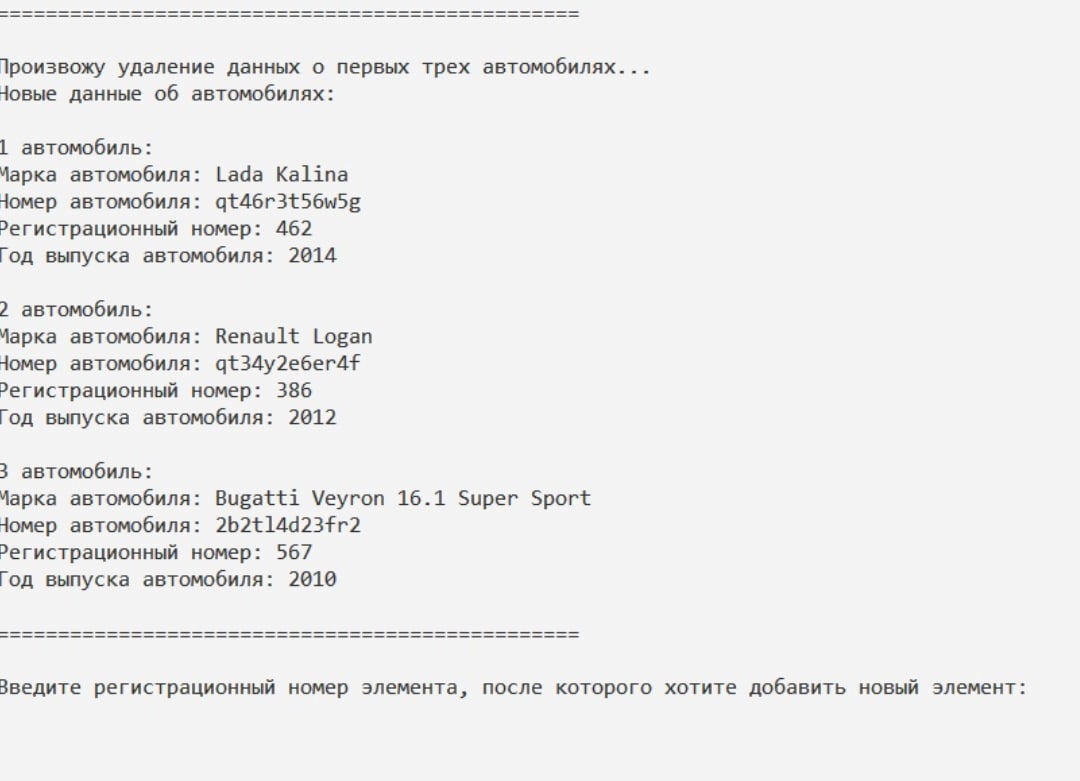
cout << endl << "================================================" << endl << endl;

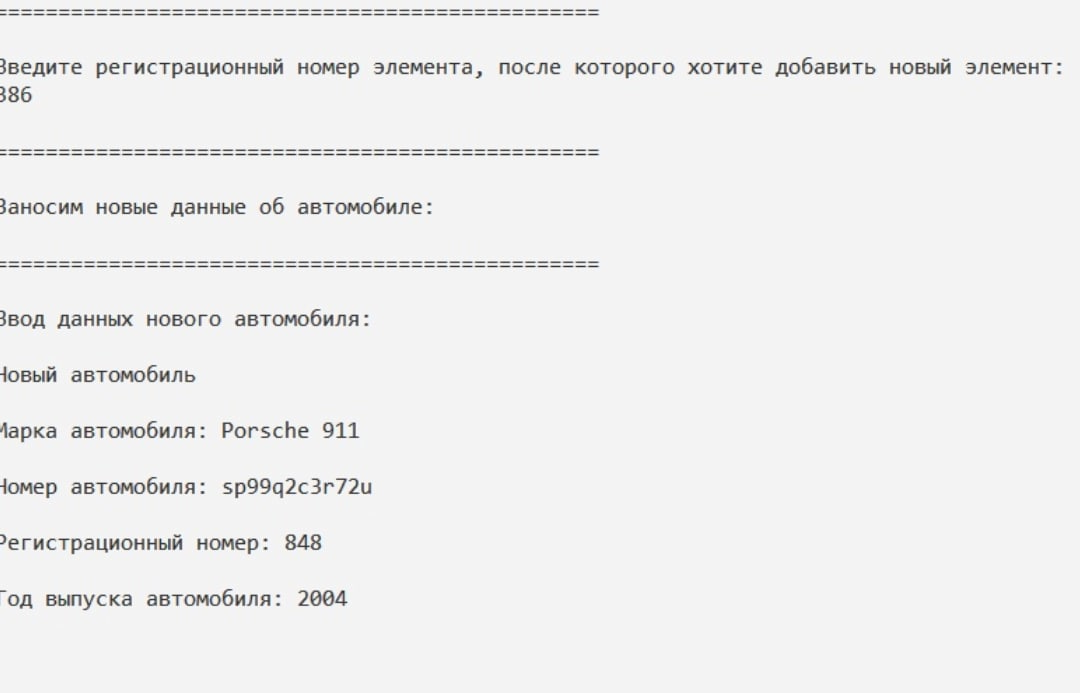
}

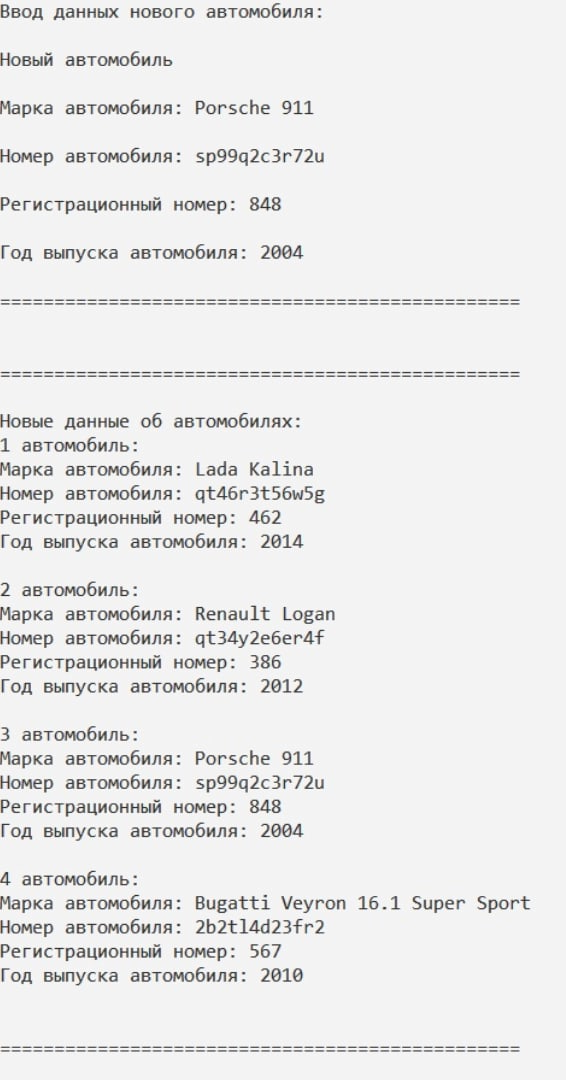
**Скриншоты результатов работы программы**

****

****

****

****

****