Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа №4

По дисциплине «Основы алгоритмизации и программирования»

На тему «Представление информации в виде структуры»

Выполнил(а):

Студент(ка) 1 курса 8 группы

Семёнов Даниил Вячеславович

Преподаватель: асс. Андронова М.В.

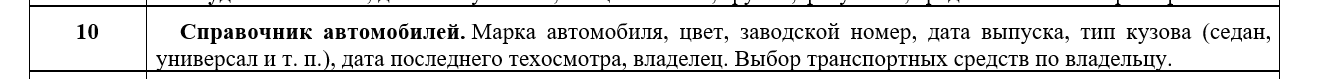
Минск, 2024

4. В соответствии со своим вариантом разработать программу для данных, приведенных в таблице ниже. Определить структурированный тип, разработать меню для работы с ***массивом структур***.

В программу должны войти функции:

* ввод элементов структуры с клавиатуры;
* вывод элементов структуры в консольное окно;
* удаление заданной структурированной переменной;
* поиск информации;
* запись информации в файл;
* чтение данных из файла.

Вариант 10



#include <iostream>

#include <fstream> // библиотека для работы с файлами

#include <string> // для работы со строками

#include <windows.h> // для корректного отображения символов

#define size 30 //подключение макроса для длины массива

using namespace std;

void input(); //прототип функции ввода

void output(); // прототип функции вывода

void del(); // прототип функции для удаления

void find(); // прототип функции поиска

void inFile(); // прототип функции записи в файл

void outFile(); // прототип функции чтения из файла

struct Book // объявление структуры

{

// обьявленние переменных структуры

string carmodel;

string color;

string carnumb;

string daterelease;

string typecar;

int datetech;

string owner;

};

Book list\_of\_cars[size]; // массив типа book для хранения информации

int current\_size=0; // текущее заполнение массива

int cars; // количество машин

int choice; // выбор

int main()

{

setlocale(LC\_ALL, "Rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//меню выбора

do

{

cout << "1 - Ввод элементов структуры с клавиатуры"<<endl;

cout << "2 - Вывод элементов структуры в консольное окно" << endl;

cout << "3 - Удаление заданной структурированной переменной" << endl;

cout << "4 - Поиск информации по владельцу" << endl;

cout << "5 - Запись информации в файл" << endl;

cout << "6 - Чтение данных из файла" << endl;

cout << "7 - Выход из программы" << endl;

cin >> choice;

// анализируем choice

switch (choice)

{

case 1:

input();

break;

case 2:

output();

break;

case 3:

del();

break;

case 4:

find();

break;

case 5:

inFile();

break;

case 6:

outFile();

break;

}

}

while (choice!=7);

return 0;

}

void input() // функция ввода

{

cout << "Введите количество автомобилей: "; cin >> cars;

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

{

if (current\_size < size) // условие для проверки наличия места

{

// заполнение информации

cout << "Информация о " << i + 1 << " автомобиле: " << endl;

cin.ignore();

cout << endl << "Марка автомобиля " << endl;

getline(cin, list\_of\_cars[current\_size].carmodel);

cout << "Цвет " << endl;

getline(cin, list\_of\_cars[current\_size].color);

cout << "Дата выпуска " << endl;

getline(cin, list\_of\_cars[current\_size].daterelease);

cout << "Заводской номер " << endl;

getline(cin, list\_of\_cars[current\_size].carnumb);

cout << "Тип кузова " << endl;

getline(cin, list\_of\_cars[current\_size].typecar);

cout << "Владелец " << endl;

getline(cin, list\_of\_cars[current\_size].owner);

cout << "Дата техосмотра " << endl;

cin >> list\_of\_cars[current\_size].datetech;

current\_size++;

}

}

}

void output() // функция вывода

{

for (int i = 0; i < current\_size; i++) // цикл вывода информации

{

cout << "Информация о " << i + 1 << " автомобиле: " << endl;

cout << endl << "Марка автомобиля " << list\_of\_cars[i].carmodel<<endl;

cout << "Цвет " << list\_of\_cars[i].color<<endl;

cout << "Дата выпуска " << list\_of\_cars[i].daterelease << endl;

cout << "Заводской номер " << list\_of\_cars[i].carnumb << endl;

cout << "Тип кузова " << list\_of\_cars[i].typecar << endl;

cout << "Владелец " << list\_of\_cars[i].owner << endl;

cout << "Дата техосмотра " << list\_of\_cars[i].datetech <<endl;

}

}

void del() // функция удаления

{

int n;

cout << "\nНомер автомобиля в каталоге, который надо удалить" << endl; cin >> n;

for (int de1 = (n - 1); de1 < current\_size; de1++)

{

list\_of\_cars[de1] = list\_of\_cars[de1 + 1];

current\_size = current\_size - 1;

}

}

void find() // функция поиска по владельцу

{

string findOwner;

cin.ignore();

bool f = true; // переменная для понимания нашли мы владельца или нет

cout << "Введите данные об владельце: ";

getline(cin, findOwner);

for (int i = 0; i < current\_size;i++)

{

if (findOwner == list\_of\_cars[i].owner)

{

cout << "Информация о " << i + 1 << " автомобиле: " << endl;

cout << endl << "Марка автомобиля " << list\_of\_cars[i].carmodel << endl;

cout << "Цвет " << list\_of\_cars[i].color << endl;

cout << "Дата выпуска " << list\_of\_cars[i].daterelease << endl;

cout << "Заводской номер " << list\_of\_cars[i].carnumb << endl;

cout << "Тип кузова " << list\_of\_cars[i].typecar << endl;

cout << "Владелец " << list\_of\_cars[i].owner << endl;

cout << "Дата техосмотра " << list\_of\_cars[i].datetech << endl;

f = false; // если нашли владельца false

}

}

if (f) // если f == true значит мы не нашли владельца

{

cout << "Информация не найдена!"<<endl;

}

}

void inFile() // функция записи в файл

{

ofstream file1("FILE1.txt");

if (file1)

{

for (int i = 0; i < current\_size; i++)

{

file1 << "Информация о " << i + 1 << " автомобиле: " << endl;

file1 << endl << "Марка автомобиля " << list\_of\_cars[i].carmodel << endl;

file1 << "Цвет " << list\_of\_cars[i].color << endl;

file1 << "Дата выпуска " << list\_of\_cars[i].daterelease << endl;

file1 << "Заводской номер " << list\_of\_cars[i].carnumb << endl;

file1 << "Тип кузова " << list\_of\_cars[i].typecar << endl;

file1 << "Владелец " << list\_of\_cars[i].owner << endl;

file1 << "Дата техосмотра " << list\_of\_cars[i].datetech << endl;

}

file1.close();

}

else

cout << "Не удалось открыть файл!"<<endl;

}

void outFile() // функция чтения из файла

{

ifstream file1("FILE1.txt");

string line;

while (getline(file1, line))

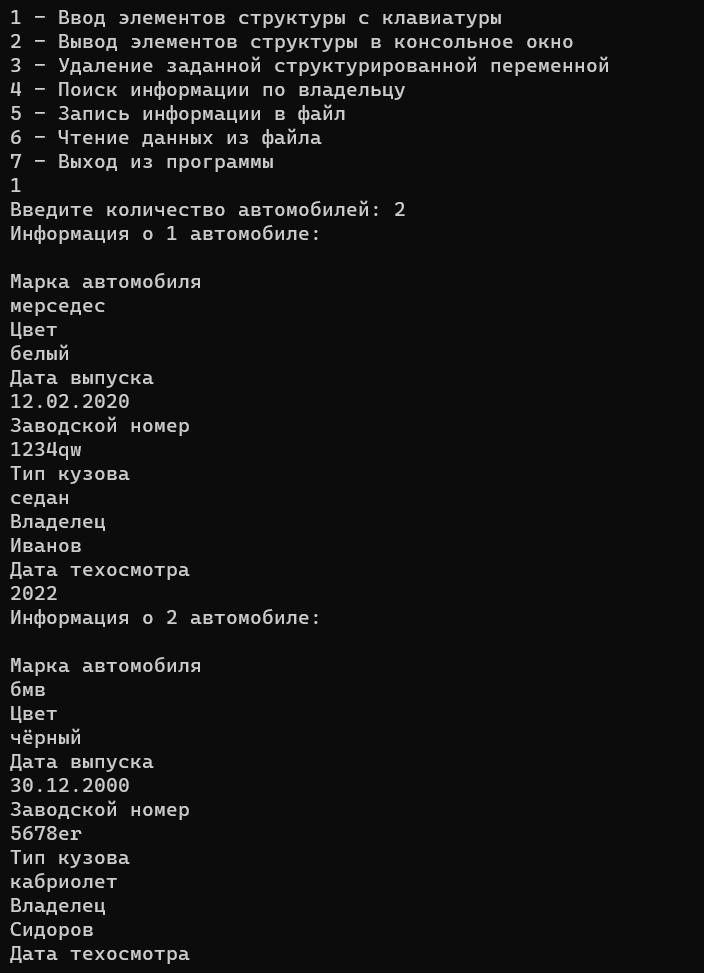
{

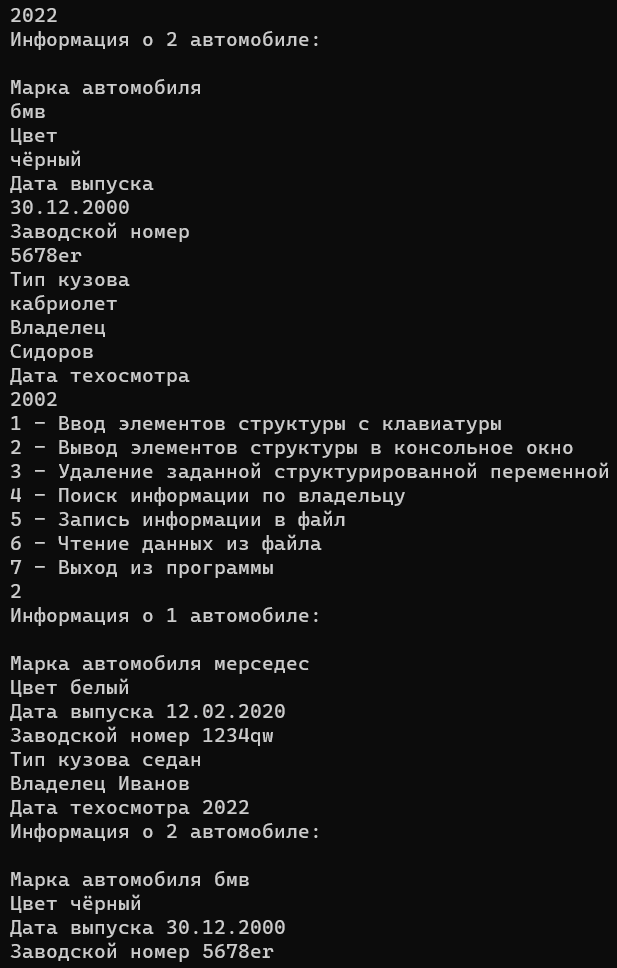
cout << line;

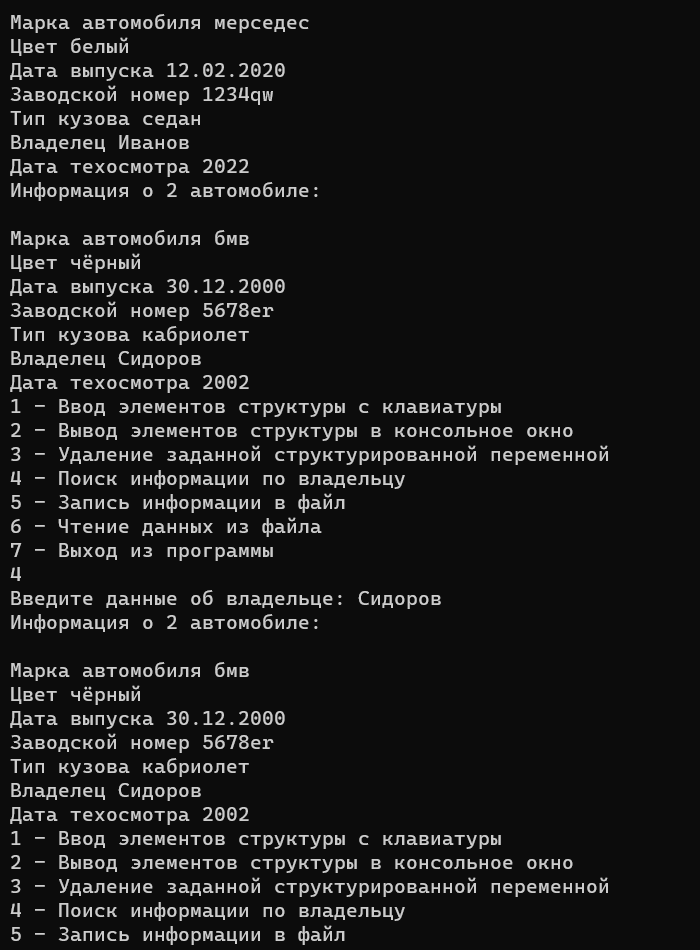
}

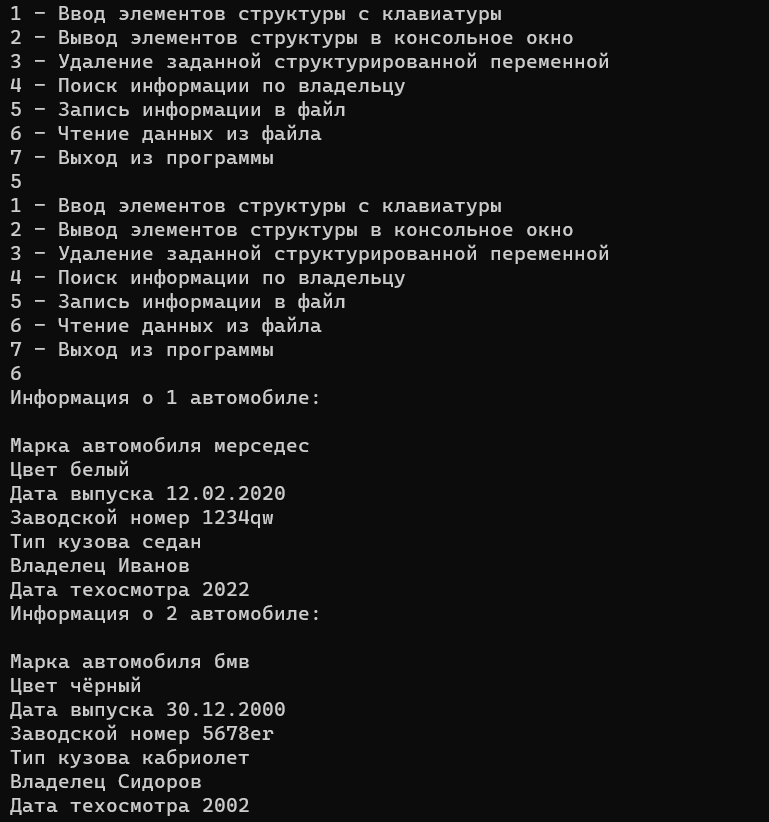
file1.close();

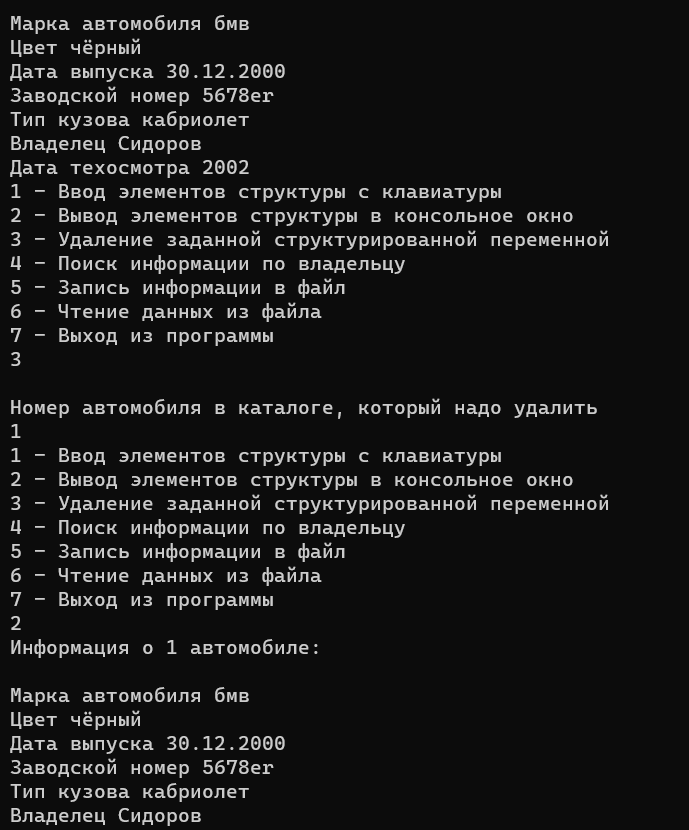
}

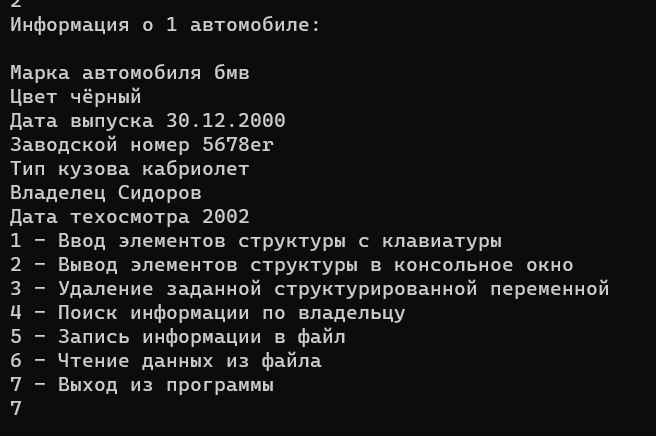


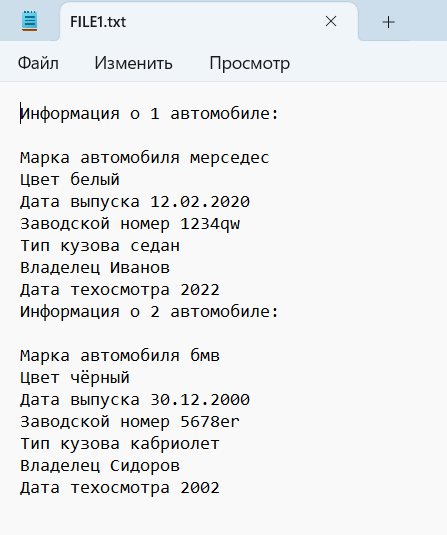




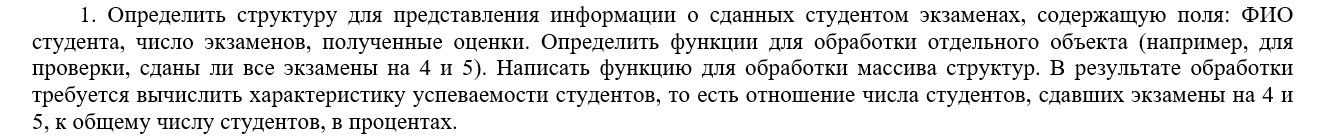








Доп. Задачи

****

#include <iostream>

#include <string>

#include <Windows.h>

#define stud 30 // максимальное значение студентов

#define size 6 // максимальное занчение массива с оценками

using namespace std;

void input(); // функция ввода

void output(); // функция вывода

void charact(); // функция для проверки сдал ли студент экзамены на 4, 5

void sec(); // функция для того чтобы узнать общую успеваемость

int choice = 0; // выбор

int current\_stud = 0; // текущее значение массива

struct Student // стркутура

{

string name;

int num\_exams;

int marks[size];

};

Student list\_of\_students[stud];

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "rus");

do

{

cout << "1 - Ввод элементов структуры с клавиатуры" << endl;

cout << "2 - Вывод элементов структуры в консольное окно" << endl;

cout << "3 - Проверка на успешность" << endl;

cout << "4 - Характеристика успеваемости" << endl;

cout << "5 - Выход из программы" << endl;

cin >> choice;

// анализируем choice

switch (choice)

{

case 1:

input();

break;

case 2:

output();

break;

case 3:

charact();

break;

case 4:

sec();

break;

}

} while (choice != 5);

return 0;

}

void input() // функция ввода

{

int numb;

cout << "Введите количесвто студентов" << endl; cin >> numb;

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

{

if (current\_stud < stud)

{

cin.ignore();

cout << endl << "Фамилия студента: " << endl;

getline(cin, list\_of\_students[current\_stud].name);

cout << "Количество экзаменов " << endl;

cin >> list\_of\_students[current\_stud].num\_exams;

cout << "Оценки " << endl;

for (int j = 0;j < list\_of\_students[current\_stud].num\_exams;j++) //цикл для записи оценок

{

cout << "Введите оценку за " << j + 1 << " экзамен: ";

cin >> list\_of\_students[current\_stud].marks[j];

cout << endl;

}

current\_stud++;

}

}

}

void output() // функция вывода

{

for (int i = 0;i < current\_stud;i++)

{

cout << endl << "Фамилия студента: " << list\_of\_students[i].name << endl;

cout << "Количество экзаменов " << list\_of\_students[i].num\_exams << endl;

cout << "Оценки: " << endl;

for (int j = 0;j < list\_of\_students[i].num\_exams;j++)

{

cout << "Введите оценку за " << j + 1 << " экзамен: "<<list\_of\_students[i].marks[j] << endl;

}

}

}

void charact() // функция для проверки сдал ли студент экзамены на 4, 5

{

int count = 0;

bool f = true;

string surname;

cout << "Введите фамилию студента: ";

cin.ignore();

getline(cin, surname);

for (int i = 0; i < current\_stud;i++)

{

if (list\_of\_students[i].name == surname)

{

for (int j = 0;j < list\_of\_students[i].num\_exams;j++)

{

if (list\_of\_students[i].marks[j] >= 4)

{

count++; // считаем оценки 4 и 5 у студента

}

}

if (count == list\_of\_students[i].num\_exams) // если их количество совпало с количеством экзаменов

{

cout << endl << "Студент сдал все экзамены на 4, 5"<<endl; // значит студент молодец

}

else

{

cout << endl << "Студент не сдал все экзамены на 4, 5"<<endl;

}

f = false;

}

}

if (f) // условие если у нас не найдено студента с такой фамилией

{

cout << "Студента с такой фамилией не найдено!" << endl;

}

}

void sec() // функция для того чтобы узнать общую успеваемость

{

float secstud=0; // переменная для хранения студентов у которых оценки выше 4

for (int i = 0;i < current\_stud;i++)

{

int count = 0;

for (int j = 0;j < list\_of\_students[i].num\_exams;j++)

{

if (list\_of\_students[i].marks[j] >= 4)

{

count++; // считаем оценки 4 и 5

}

}

if (count == list\_of\_students[i].num\_exams)

{

secstud++; // увеличиваем значение

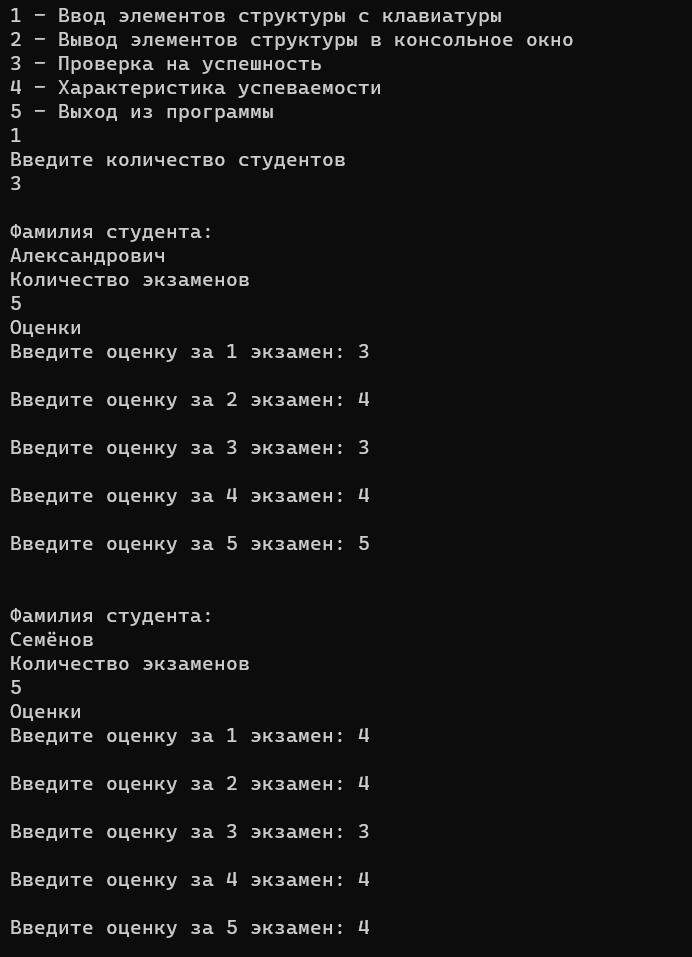
}

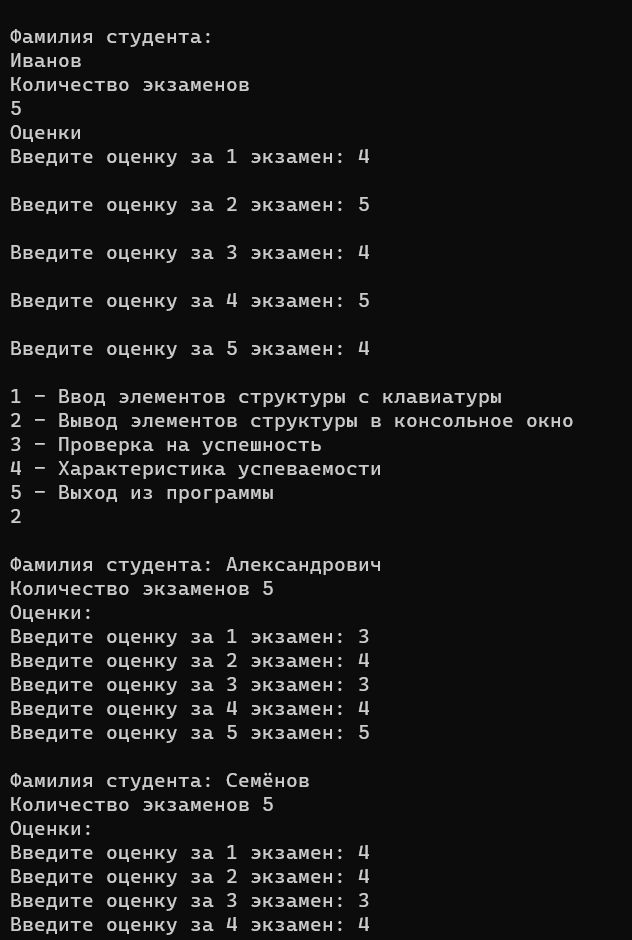
}

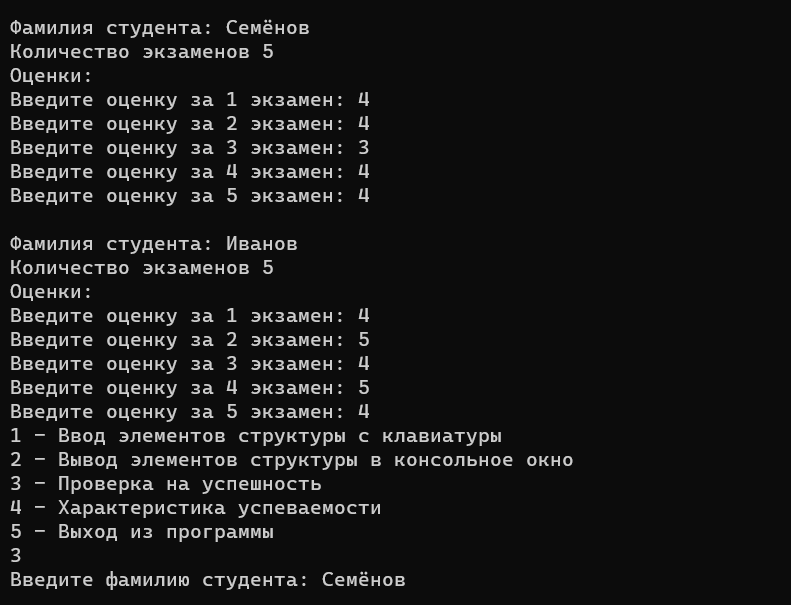
float devide = secstud / current\_stud\*100; // считаем наших студентов в процентах

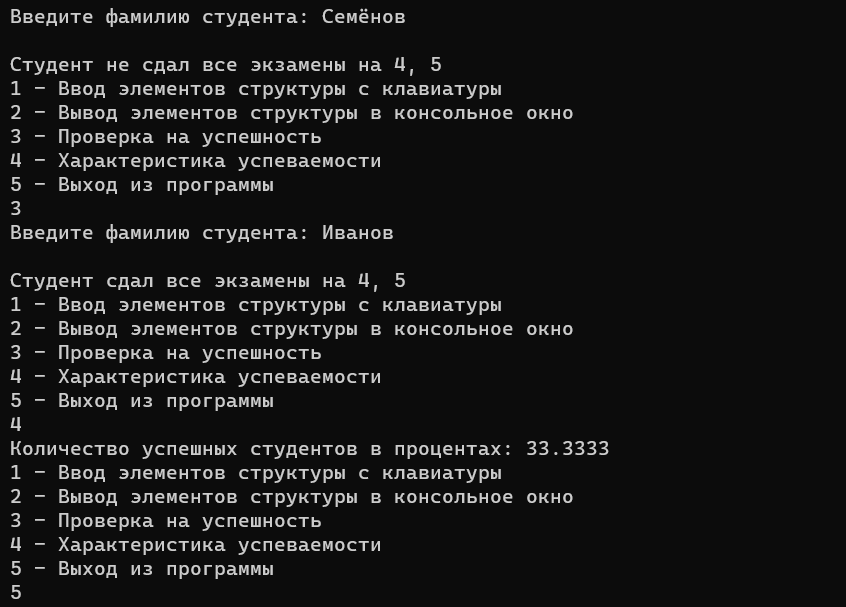
cout << "Количество успешных студентов в процентах: " << devide << endl;

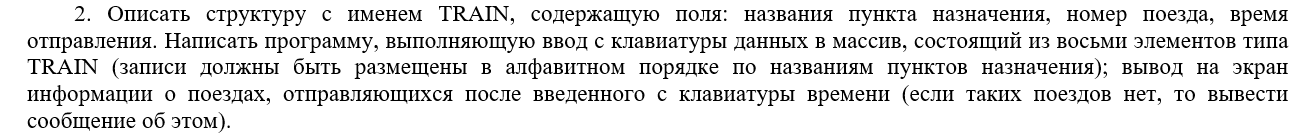
}



****

****

****

****

#include <iostream>

#include <string>

#include <Windows.h>

#define size 8

using namespace std;

void input(); // функция для ввода

void output(); // функция для вывода

void sortoftime(); // функция для вывода поездов отправляющихся по времени после введённого числа с клавиатуры

struct TRAIN

{

string destination;

int number;

string time;

};

TRAIN trains[size];

int choice;

int main()

{

setlocale(LC\_ALL, "rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//меню выбора

do

{

cout << "1 - Ввод элементов структуры с клавиатуры" << endl;

cout << "2 - Вывод элементов структуры в консольное окно" << endl;

cout << "3 - Вывод элементов после введённого времени" << endl;

cout << "4 - Выход из программы" << endl;

cin >> choice;

// анализируем choice

switch (choice)

{

case 1:

input();

break;

case 2:

output();

break;

case 3:

sortoftime();

break;

}

} while (choice != 4);

return 0;

}

void input() // функция для ввода

{

for (int i = 0;i < 8;i++)

{

if (i < size)

{

// заполнение информации

cin.ignore();

cout << "Информация о " << i + 1 << " поезде: " << endl;

cout << endl << "Пункт назначения " << endl;

getline(cin, trains[i].destination);

cout << "Время отправления " << endl;

getline(cin, trains[i].time);

cout << "Номер поезда " << endl;

cin >> trains[i].number;

}

}

for (int i = 0;i < 8;i++) // сортируем массив по пунктам назначения

{

for (int j = i + 1;j < 8;j++)

{

if(trains[i].destination > trains[j].destination)

{

TRAIN buff = trains[i];

trains[i] = trains[j];

trains[j] = buff;

}

}

}

}

void output() // функция для вывода

{

for (int i = 0;i < 8;i++)

{

cout << endl << "Пункт назначения " << trains[i].destination << endl;

cout << "Номер поезда " << trains[i].number << endl;

cout << "Время отправления " << trains[i].time << endl;

}

}

void sortoftime() // функция для вывода поездов отправляющихся по времени после введённого числа с клавиатуры

{

string time;

cout << "Введите время отправления: ";

cin.ignore();

getline(cin, time);

bool f=true;

for (int i = 0;i < 8;i++)

{

if (trains[i].time > time)

{

cout << endl << "Пункт назначения " << trains[i].destination << endl;

cout << "Номер поезда " << trains[i].number << endl;

cout << "Время отправления " << trains[i].time << endl;

f = false;

}

}

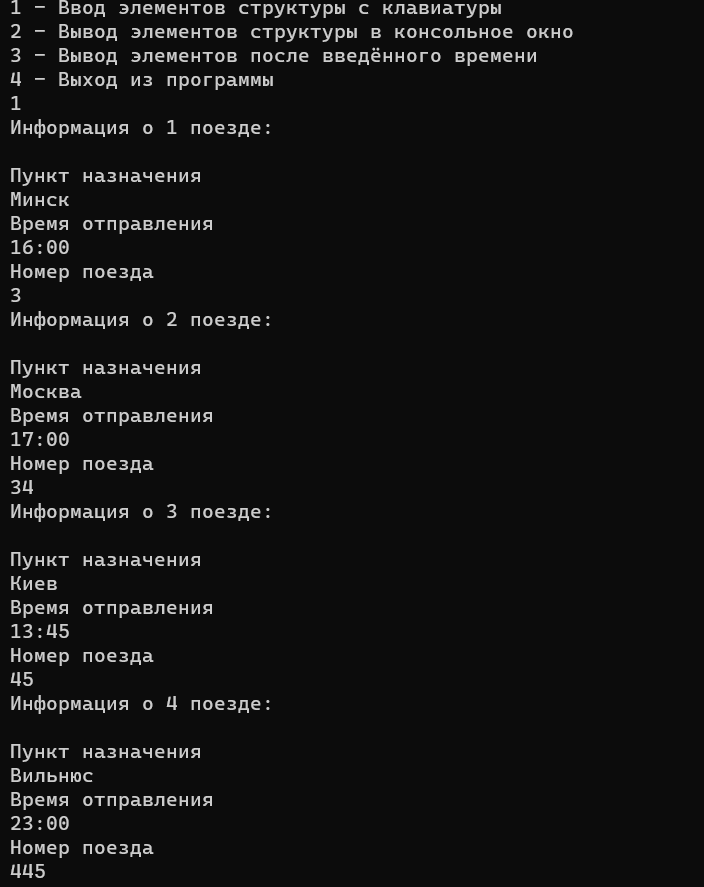
if (f)

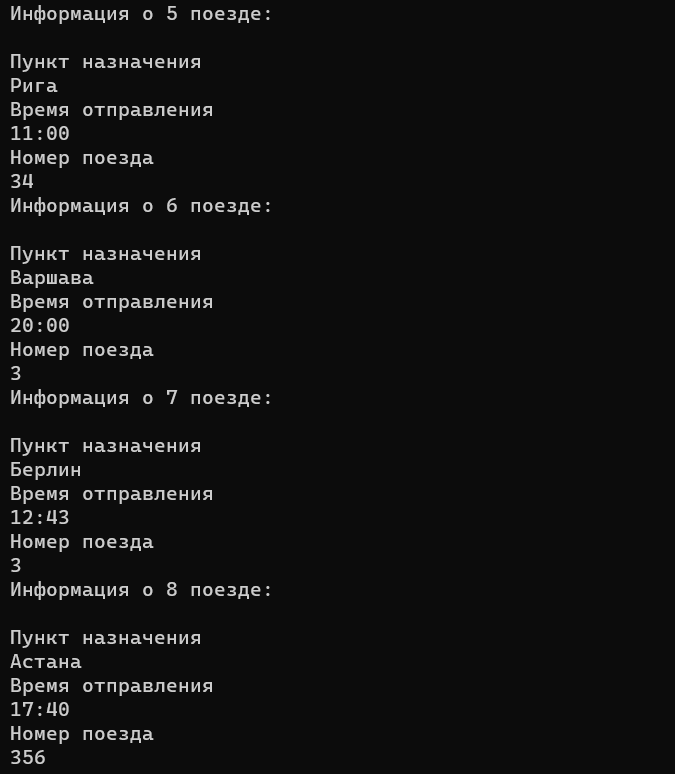
{

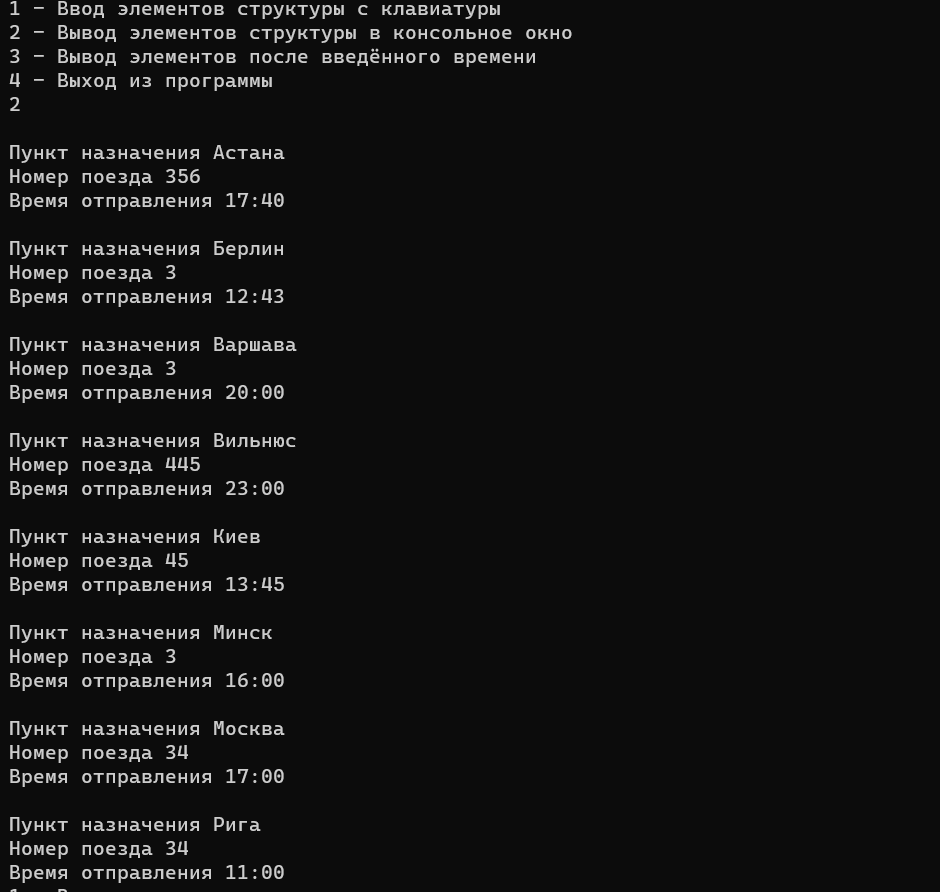
cout << "Поездов не найдено" << endl;

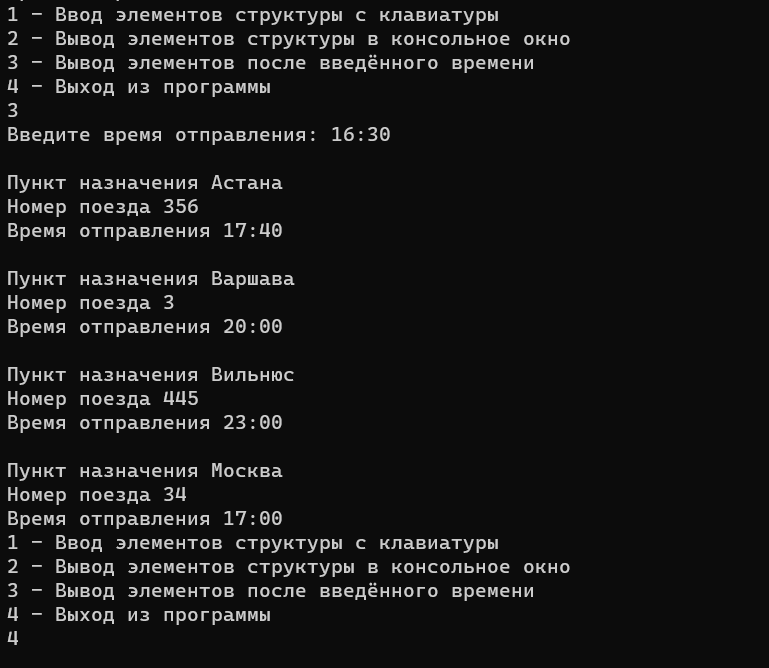
}

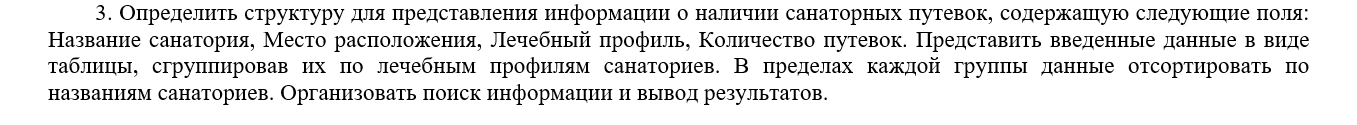
}

****

****

****

****

****

#include <iostream>

#include <string>

#include <Windows.h>

#define size 10

using namespace std;

void input(); // функция для ввода

void output(); // функция для вывода

void search(); // функция для поиска

struct Sanatorium

{

string name;

string location;

string profile;

int tickets;

};

Sanatorium sanat[size];

Sanatorium buff;

int current\_size = 0;

int choice;

int main()

{

setlocale(LC\_ALL, "rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//меню выбора

do

{

cout << "1 - Ввод элементов структуры с клавиатуры" << endl;

cout << "2 - Вывод элементов структуры в консольное окно" << endl;

cout << "3 - Поиск элементов" << endl;

cout << "4 - Выход из программы" << endl;

cin >> choice;

// анализируем choice

switch (choice)

{

case 1:

input();

break;

case 2:

output();

break;

case 3:

search();

break;

}

} while (choice != 4);

return 0;

}

void input()

{

int s;

cout << "Сколько санаториев вы хотите ввести? ";

cin >> s;

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

{

// заполнение информации

if (current\_size < size)

{

cin.ignore();

cout << "Информация о " << i + 1 << " санатории: " << endl;

cout << endl << "Название санатория " << endl;

getline(cin, sanat[i].name);

cout << "Место расположения " << endl;

getline(cin, sanat[i].location);

cout << "Лечебный профиль " << endl;

getline(cin, sanat[i].profile);

cout << "Количество путёвок " << endl;

cin >> sanat[i].tickets;

current\_size++;

}

}

}

void output()

{

for (int i = 0; i < current\_size; i++) // сортируем

{

for (int j = i + 1; j < current\_size; j++)

{

if (sanat[i].profile > sanat[j].profile || (sanat[i].profile == sanat[j].profile && sanat[i].name > sanat[j].name))

{

buff = sanat[i];

sanat[i] = sanat[j];

sanat[j] = buff;

}

}

}

for (int i = 0; i < current\_size; i++) // выводим

{

cout << "Название: " << sanat[i].name << ", Место: " << sanat[i].location << ", Профиль: " << sanat[i].profile << ", Количество путёвок: " << sanat[i].tickets << endl;

}

}

void search() // функция для поиска

{

string profile;

bool f=true;

cout << "Введите профиль для поиска: ";

cin.ignore();

getline(cin, profile);

for (int i = 0; i < current\_size; i++)

{

if (sanat[i].profile == profile)

{

cout << "Название: " << sanat[i].name << ", Место: " << sanat[i].location << ", Профиль: " << sanat[i].profile << ", Количество путёвок: " << sanat[i].tickets << endl;

f = false;

}

}

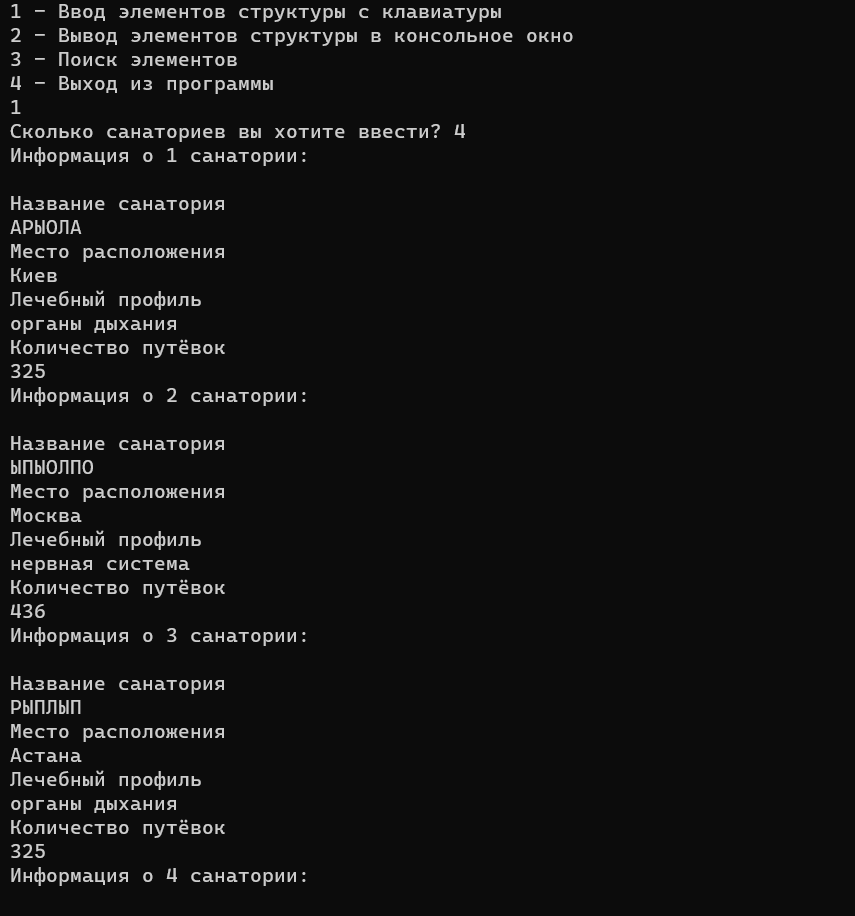
if (f)

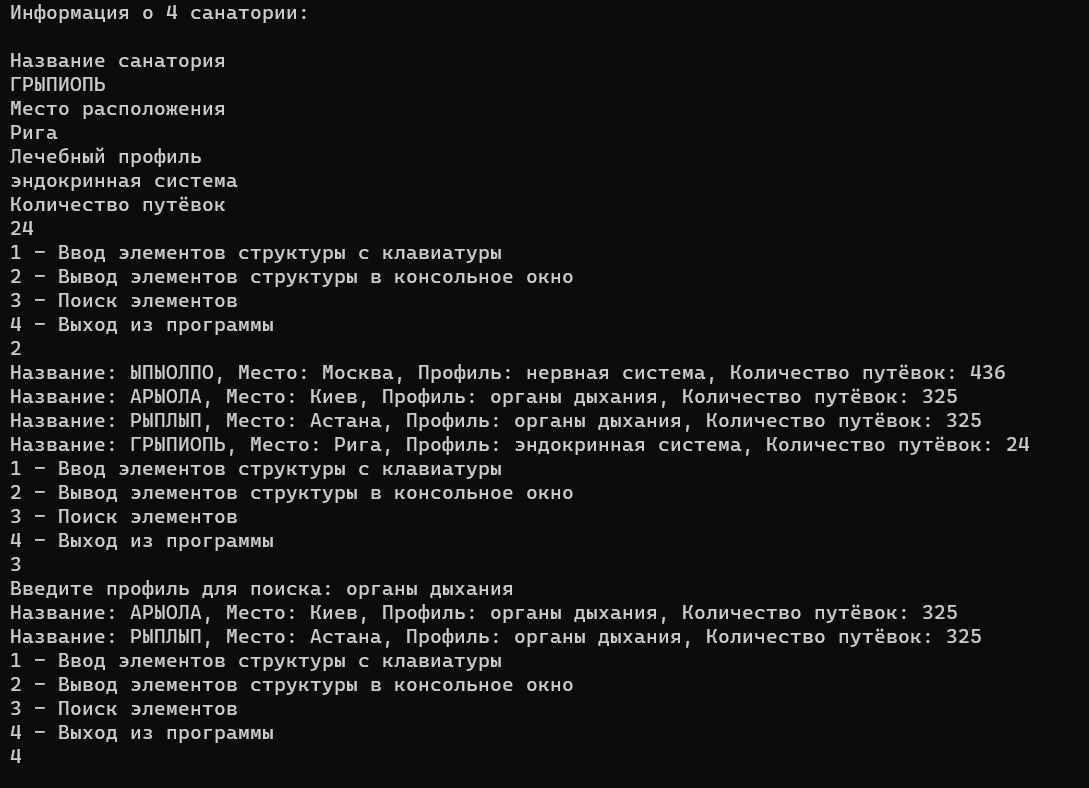
{

cout << "Санаторий не найден!" << endl;

}

}

****

****