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:

Федорович Вадим

Студент 1 курса 8 группы

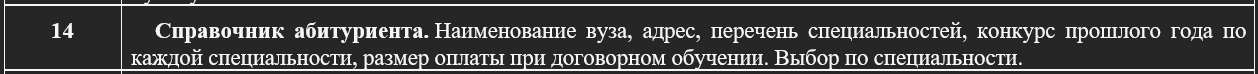
Преподаватель: асс. Андронова М.В.

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.



include<iostream>

#include<Windows.h>

#include<stdio.h>

#include <string>

#include<fstream>

using namespace std;

#define size 55

void enter();// Функция ввода

void out(); // Функция вывода

void delet();// Функция удвления

void found();// Функция поиска

void write\_file();// Функция записи в файл

void read\_file();// Функция чтения фала

void choice\_specialties();//

struct applicant { //Структура

string university;

string location;

string specialties;

string contest;

string cost;

};

struct applicant applicants\_guide[size];

struct applicant bad;

int current\_size = 0; //Для подсчета количества введенного

int choice;

int main() {

SetConsoleCP(1251);

setlocale(LC\_CTYPE, "Russian");

cout << "Введите:" << endl;

cout << "1 - Для ввода новой записи" << endl;

cout << "2 - Для вывода записи(ей)" << endl;

cout << "3 - Для удаления записи" << endl;

cout << "4 - Поиск записей" << endl;

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

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

cout << "7 - Поиск по специальности" << endl;

cout << "8 - Для выхода" << endl;

cin >> choice;

do

{

switch (choice) {

case 1: enter(); break;

case 2: out(); break;

case 3: delet(); break;

case 4: found(); break;

case 5: write\_file(); break;

case 6: read\_file(); break;

case 7: choice\_specialties(); break;

}

} while (choice != 8);

}

void enter() {

cout << "Ввод информации" << endl;

string choice\_str;

do {

if (current\_size < size) {

cout << "Строка номер " << current\_size + 1 << endl;

cout << "Наименование вуза: ";

cin.ignore(); // очистка буфера ввода

getline(cin, applicants\_guide[current\_size].university);

cout << "Адрес: ";

getline(cin, applicants\_guide[current\_size].location);

cout << "Перечень специальностей: ";

getline(cin, applicants\_guide[current\_size].specialties);

cout << "Конкурс прошлого года по каждой специальности: ";

getline(cin, applicants\_guide[current\_size].contest);

cout << "Размер оплаты при договорном обучении: ";

getline(cin, applicants\_guide[current\_size].cost);

current\_size++;

}

else {

cout << "Введено максимальное количество строк" << endl;

break;

}

cout << "Продолжить ввод? (1 - да, 0 - нет): ";

cin >> choice\_str;

// Проверяем, введено ли "0" для завершения ввода

} while (choice\_str == "1");

cout << "Что дальше?" << endl;

cin >> choice;

}

void out() {

int answer, number;

cout << "1 - Вывод всех строк" << endl;

cout << "2 - Вывод одной строки" << endl;

cin >> answer;

if (answer == 2) {

cout << "Номер выводимой строки? " << endl;

cin >> number;

cout << endl << "Наименование вуза " << endl;

cout << applicants\_guide[number - 1].university;

cout << "Адрес " << endl;

cout << applicants\_guide[number - 1].location;

cout << "Перечень специальностей " << endl;

cout << applicants\_guide[number - 1].specialties;

cout << "конкурс прошлого года по каждой специальности " << endl;

cout << applicants\_guide[number - 1].contest;

cout << "размер оплаты при договорном обучении " << endl;

cout << applicants\_guide[number - 1].cost;

}

if (answer == 1) {

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

cout << endl << "Наименование вуза: ";

cout << applicants\_guide[i].university << endl;

cout << "Адрес: ";

cout << applicants\_guide[i].location << endl;

cout << "Перечень специальностей: ";

cout << applicants\_guide[i].specialties << endl;

cout << "конкурс прошлого года по каждой специальности: ";

cout << applicants\_guide[i].contest << endl;

cout << "размер оплаты при договорном обучении: ";

cout << applicants\_guide[i].cost << endl;

}

cout << "Что дальше?" << endl;

cin >> choice;

}

}

void delet() {

int number, digit;

cout << "1 - Хочу удалить все строки" << endl;

cout << "2 - Хочу удалить одну строку" << endl;

cin >> number;

if (number == 2) {

cout << "Какую строку хотите удалить? " << endl;

cin >> digit;

applicants\_guide[digit] = bad; //Присваивание пустой структуры

cout << "Строка удалена " << endl;

}

if (number == 1) {

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

applicants\_guide[i] = bad; //Присваивание пустой структуры

}

cout << "Строки удалены " << endl;

}

cout << "Что дальше?" << endl;

cin >> choice;

}

void found() {

int count;

cout << "Что бы вы хотели найти? " << endl;

cout << "1 - Наименование вузов" << endl;

cout << "2 - Адреса" << endl;

cout << "3 - Перечни специальностей" << endl;

cout << "4 - Конкурс прошлого года по каждой специальности" << endl;

cout << "5 - Размер оплат при договорном обучении" << endl;

cin >> count;

if (count < 1 || count > 5) {

cout << "Неккоректный ответ. " << endl;

}

else {

if (count == 1) {

cout << endl << "Наименование вуза " << endl;

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

cout << applicants\_guide[i].university << endl;

}

}

if (count == 2) {

cout << "Адреса " << endl;

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

cout << applicants\_guide[i].location << endl;

}

}

if (count == 3) {

cout << "Перечень специальностей " << endl;

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

cout << applicants\_guide[i].specialties << endl;

}

}

if (count == 4) {

cout << "Конкурс прошлого года по каждой специальности " << endl;

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

cout << applicants\_guide[i].contest << endl;

}

}

if (count == 5) {

cout << "Размеры оплаты при договорном обучении: " << endl;

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

cout << applicants\_guide[i].cost << endl;

}

}

}

cout << "Что дальше?" << endl;

cin >> choice;

}

void write\_file() {

ofstream file\_write("applicant.txt");

int num;

cout << "Запись информации в файл ..." << endl;

cout << "Что хотите записать?" << endl;

cout << "1 - Все" << endl;

cout << "2 - Одну строку" << endl;

cin >> num;

if (num != 1 && num != 2) {

cout << "Некорректный ввод" << endl;

return; // Выходим из функции, так как введен некорректный номер

}

if (num == 1) {

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

file\_write << "Наименование вуза: " << applicants\_guide[i].university << endl;

file\_write << "Адрес: " << applicants\_guide[i].location << endl;

file\_write << "Перечень специальностей: " << applicants\_guide[i].specialties << endl;

file\_write << "Конкурс прошлого года по каждой специальности: " << applicants\_guide[i].contest << endl;

file\_write << "Размер оплаты при договорном обучении: " << applicants\_guide[i].cost << endl;

file\_write << endl;

cout << "Строки записаны." << endl;

}

}

if (num == 2) {

int g;

cout << "Какую строку хотите записать? " << endl;

cin >> g;

if (g >= 1 && g <= current\_size) {

file\_write << "Наименование вуза: " << applicants\_guide[g - 1].university << endl;

file\_write << "Адрес: " << applicants\_guide[g - 1].location << endl;

file\_write << "Перечень специальностей: " << applicants\_guide[g - 1].specialties << endl;

file\_write << "Конкурс прошлого года по каждой специальности: " << applicants\_guide[g - 1].contest << endl;

file\_write << "Размер оплаты при договорном обучении: " << applicants\_guide[g - 1].cost << endl;

file\_write << endl;

cout << "Строки записаны." << endl;

}

else {

cout << "Некорректный номер строки" << endl;

}

}

file\_write.close();

cout << "Что дальше?" << endl;

cin >> choice;

}

void read\_file() {

ifstream file\_read("applicant.txt");

int num;

string buffer;

cout << "1 - Прочесть весь файл " << endl;

cout << "2 - Прочесть один справочник абитурьента " << endl;

cin >> num; cout << endl;

if (num != 1 && num != 2) {

cout << "Некорректный ввод" << endl;

return; // Выходим из функции, так как введен некорректный номер

}

if (num == 1) {

int count = 1;

while (getline(file\_read, buffer)) {

cout << buffer << endl; // Выводим содержимое каждой строки файла

count++;

if (count == 6) {

cout << endl << endl;

}

}

}

if (num == 2) {

int coin, count = 0;

bool flag = true;

int near\_coin; int helper = 6;

cout << "Номер какого справочника вы бы хотели прочесть?" << endl;

cin >> coin;

near\_coin = abs(1 - coin)\*6;

while (getline(file\_read, buffer)) {

near\_coin--;

if (coin == 1) {

flag = false;

helper--;

cout << buffer << endl;

if (helper == 6) {

break;

}

}

if (near\_coin <= 0 && helper != 0 && flag == true) {

cout << buffer << endl; // Выводим содержимое каждой строки файла

helper--;

}

}

}

cout << "Что дальше?" << endl;

cin >> choice;

}

void choice\_specialties() {

string spec;

cout << "Введите специальность" << endl;

cin >> spec;

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

if (applicants\_guide[i].specialties == spec) { //Проверка по специальности

cout << "Наименование вуза: " << applicants\_guide[i].university << endl;

cout << "Адрес: " << applicants\_guide[i].location << endl;

cout << "Перечень специальностей: " << applicants\_guide[i].specialties << endl;

cout << "Конкурс прошлого года по каждой специальности: " << applicants\_guide[i].contest << endl;

cout << "Размер оплаты при договорном обучении: " << applicants\_guide[i].cost << endl;

cout << endl;

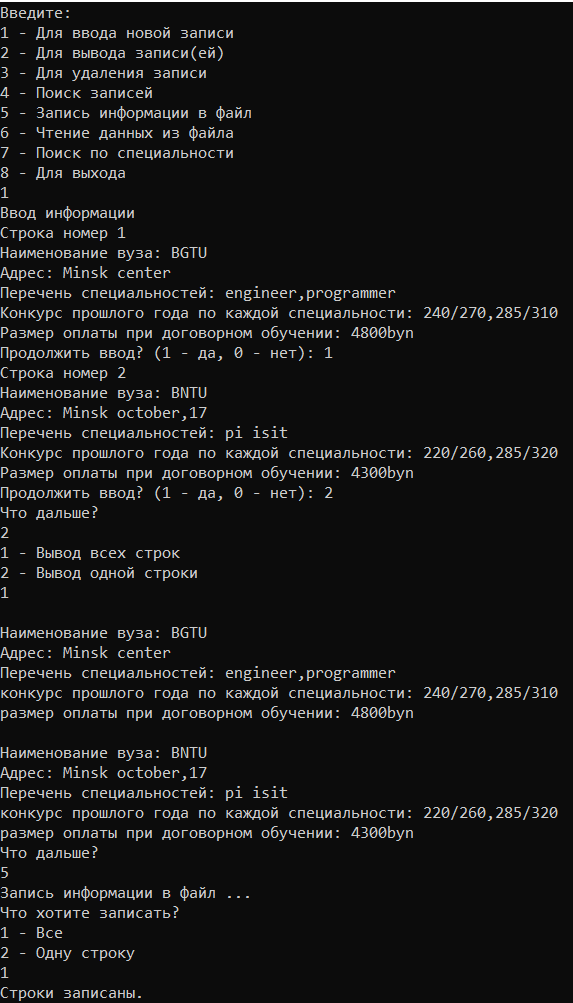
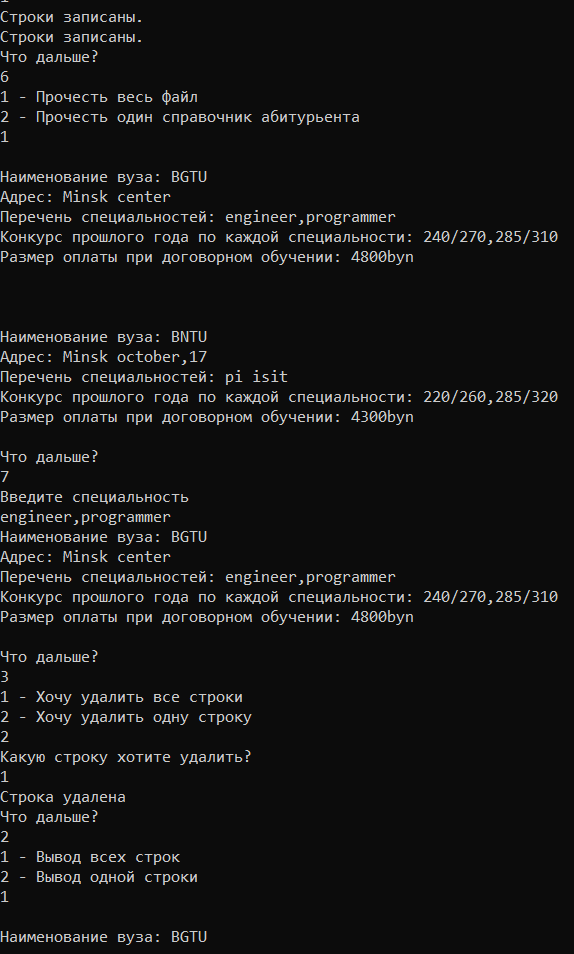
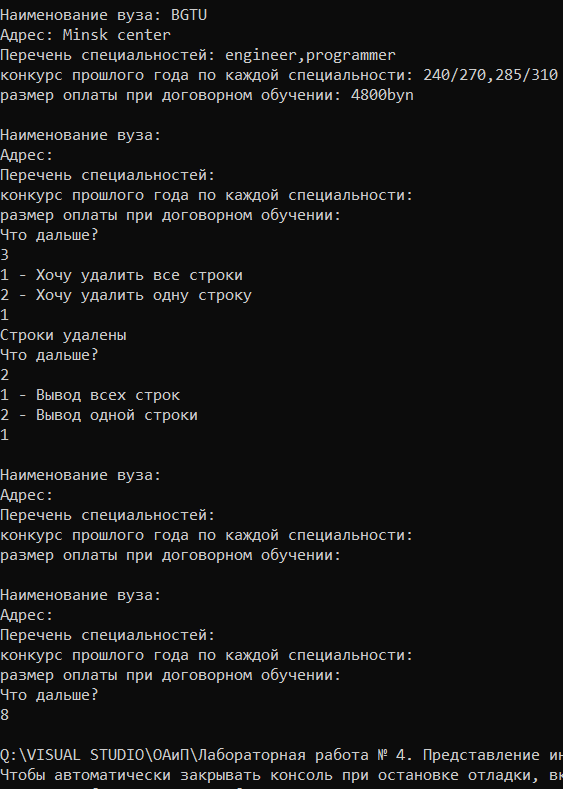
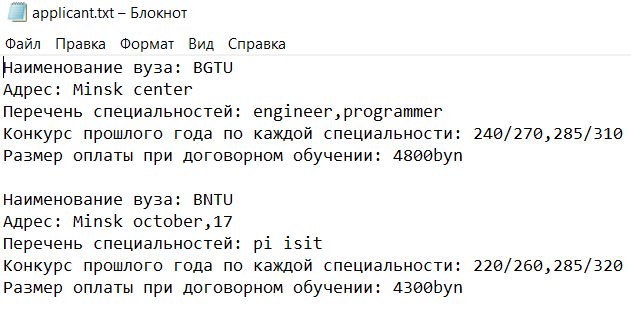
}

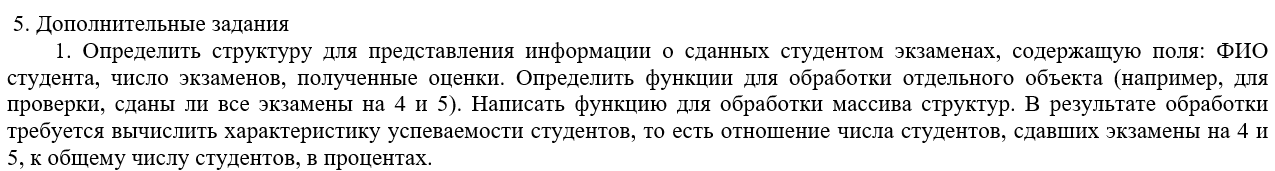
}

cout << "Что дальше?" << endl;

cin >> choice;

}



#include<iostream>

#include<Windows.h>

#include<stdio.h>

#include <string>

#include<fstream>

using namespace std;

#define size 15 //Длинна массива структуры

#define Max\_rating 10

struct students {

string FIO;

int count\_of\_exams;

int rating[Max\_rating];

};

struct students students\_mass[size];

int enter\_size = 0; //Для подсчета количества введенного массива

int choice;

void enter(); //Функция ввода

void test\_rating();// Функция проверки оценок

void perfomance\_calculation();// Функция характеристики успеваемости студентов

int main() {

SetConsoleCP(1251);

setlocale(LC\_ALL, "rus");

cout << "Введите: " << endl;

cout << "1 - Ввод данных" << endl;

cout << "2 - Проверка на оценки" << endl;

cout << "3 - Вычислить характеристику успеваемости студентов" << endl;

cout << "5 - Выход" << endl;

cin >> choice;

do

{

switch (choice) {

case 1: enter(); break;

case 2: test\_rating(); break;

case 3: perfomance\_calculation(); break;

}

}

while (choice != 5);

}

void enter() {

int answer;

cout << "Ввод информации" << endl;

do {

if (enter\_size < size) {

cout << enter\_size+1 <<" Студент " << endl;

cout << "Введите ФИО студента: " << endl;

cin.ignore();

getline(cin, students\_mass[enter\_size].FIO);

cout << "Введите кол-во экзаменов: ";

cin >> students\_mass[enter\_size].count\_of\_exams;

cout << "Введите оценки";

for (int i = 0; i < students\_mass[enter\_size].count\_of\_exams; ++i) {

cin >> students\_mass[enter\_size].rating[i];

}

enter\_size++;

}

else {

cout << "Максимум места достигнут" << endl;

break;

}

cout << "Продолжить ввод? (1 - да, 0 - нет): ";

cin >> answer;

} while (answer == 1);

cout << "Что дальше?" << endl;

cin >> choice;

}

void test\_rating() {

int number; int answer;

bool flag\_exams;

do{

flag\_exams = true;

cout << "Рейтинг какого студента вы бы хотели увидеть?" << endl;

cin >> number;

for (int i = 0; i < students\_mass[number - 1].count\_of\_exams; i++) {

if (students\_mass[number-1].rating[i] < 4) {

flag\_exams = false;

}

}

if (flag\_exams == true) {

cout << "Этот студент сдал все на отлично" << endl;

}

else if (flag\_exams == false) {

cout << "У этого студента есть оценки ниже 4 и 5" << endl;

}

cout << "Продолжить ввод? (1 - да, 0 - нет): ";

cin >> answer;

} while (answer == 1);

cout << "Что дальше?" << endl;

cin >> choice;

}

void perfomance\_calculation() { //

bool flag;

int clever\_students = 0;

int students; double percent;

students = enter\_size; // количество введеных студентов

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

flag = true;

for (int g = 0; g < students\_mass[i].count\_of\_exams; g++) { // смена массивов

if (students\_mass[i].rating[g] < 4) { // смена оценок

flag = false;

}

}

if (flag == true) {

clever\_students++;

}

}

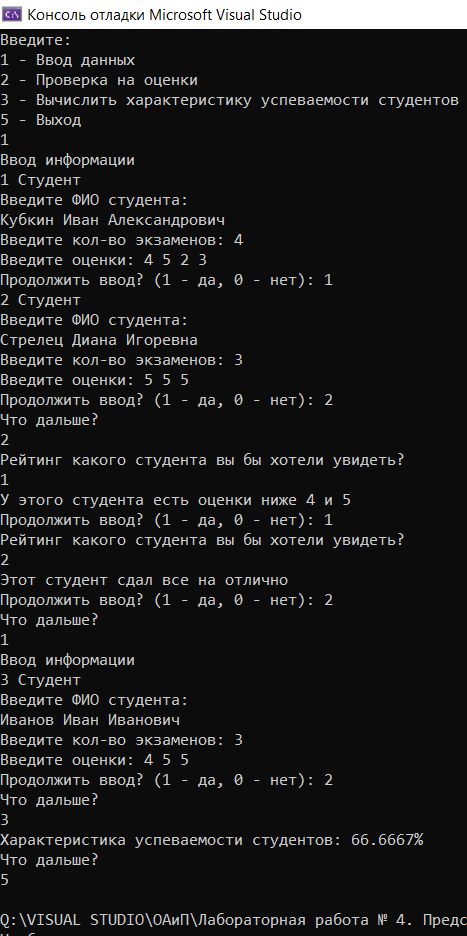
percent = (static\_cast<double>(clever\_students) / students) \* 100;

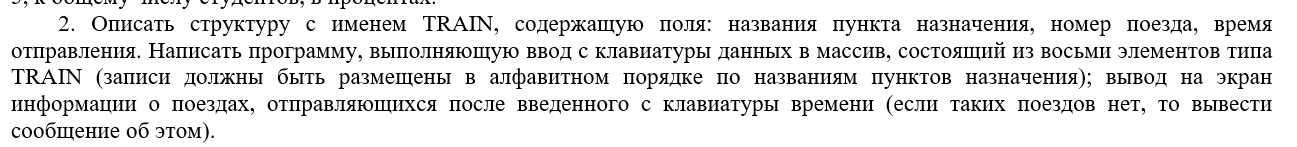
cout << "Характеристика успеваемости студентов: " << percent << "%" << endl;

cout << "Что дальше?" << endl;

cin >> choice;

}





#include<iostream>

#include<Windows.h>

#include<stdio.h>

using namespace std;

#define size 8

struct TRAIN {

char name[20];

int number;

int time;

};

TRAIN train\_mass[size];

int choice;

int now\_size = 0;

void enter();

void out();

void sort();

void all\_out();

int main() {

setlocale(LC\_ALL, "rus");

SetConsoleCP(1251);

cout << "Введите: " << endl;

cout << "1 - Ввод данных" << endl;

cout << "2 - вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времен" << endl;

cout << "3 - Сортировка станций по алфавиту" << endl;

cout << "4 - Вывод всей информации" << endl;

cout << "5 - Выход" << endl;

cin >> choice;

do {

switch (choice) {

case 1: enter(); break;

case 2: out(); break;

case 3: sort(); break;

case 4: all\_out(); break;

}

} while (choice != 5);

}

void enter() {

int answer;

do{

if (now\_size >= size) { // проверка на место

cout << "Ошибка: массив поездов заполнен." << endl;

return;

}

cout << "Введите название пункта назначения: ";

cin.ignore(); // Чистка буфера

cin.getline(train\_mass[now\_size].name, 20);

cout << "Введите номер поезда: ";

cin >> train\_mass[now\_size].number;

cout << "Введите время отправления: ";

cin >> train\_mass[now\_size].time;

now\_size++;

cout << "Продолжить ввод? (1 - да, 0 - нет): ";

cin >> answer;

} while (answer == 1);

cout << "Что дальше?" << endl;

cin >> choice;

}

void out() { //Функция

int enter\_time;

int hours;

int min;

bool flag = false;

cout << "Введите время: " << endl;

cin >> enter\_time;

hours = enter\_time / 100;

min = enter\_time % 100;

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

if ( (hours\*60 +min) < ((train\_mass[i].time/100)\*60) + (train\_mass[i].time%100)) {

cout << "Имя Станции: ";

cout << train\_mass[i].name << endl;

cout << "Номер поезда: ";

cout << train\_mass[i].number << endl;

cout << "Время отправления: ";

cout << train\_mass[i].time << endl;

cout << endl;

flag = true;

}

if (flag == false) {

cout << "Таких поездов нету" << endl;

}

}

cout << "Что дальше?" << endl;

cin >> choice;

}

void sort() {// Сортировка

cout << "Сортировка по алфавиту..." << endl;

for (int i = 0; i < now\_size - 1; i++) {

for (int j = 0; j < now\_size - i - 1; j++) {

if (strcmp(train\_mass[j].name, train\_mass[j + 1].name) > 0) {

swap(train\_mass[j], train\_mass[j + 1]);

}

}

}

cout << "Что дальше?" << endl;

cin >> choice;

}

void all\_out() {// Вывод всего записанного

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

cout << "Имя Станции: ";

cout << train\_mass[i].name << endl;

cout << "Номер поезда: ";

cout << train\_mass[i].number << endl;

cout << "Время отправления: ";

cout << train\_mass[i].time << endl;

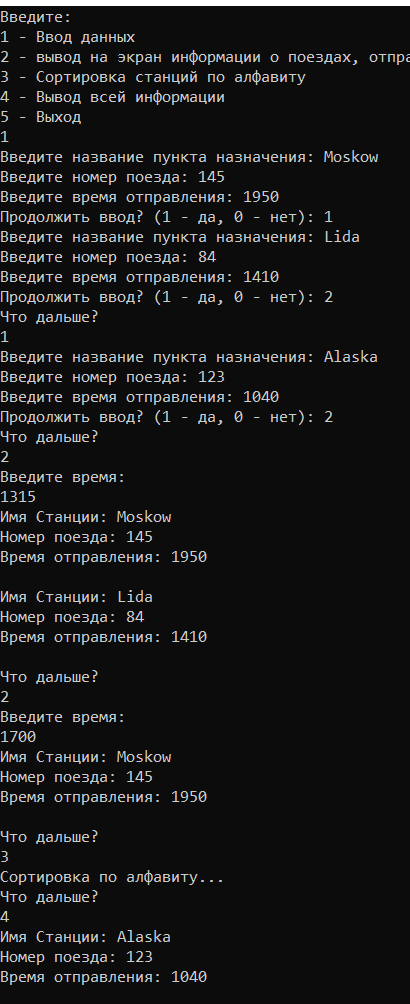
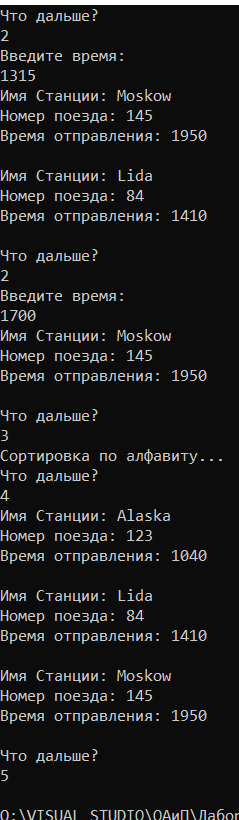
cout << endl;

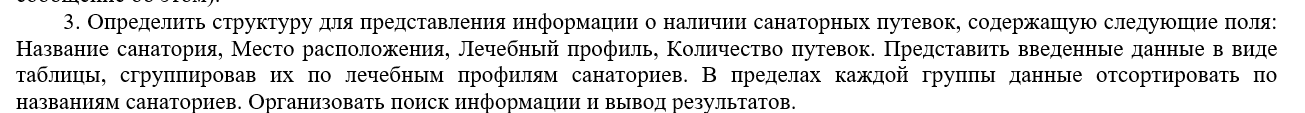
}

cout << "Что дальше?" << endl;

cin >> choice;

}



#include<iostream>

#include<Windows.h>

#include<stdio.h>

#include <string>

#include<fstream>

using namespace std;

#define size 8

struct healing {

string name;

string place;

char healing[50];

int travel\_packages;

};

healing health\_center[size];

int choice;

int now\_size = 0;

void enter();

void all\_out();

void found();

int main() {

setlocale(LC\_ALL, "rus");

SetConsoleCP(1251);

cout << "Введите: " << endl;

cout << "1 - Ввод данных" << endl;

cout << "2 - Вывод всей информации" << endl;

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

cout << "5 - Выход" << endl;

cin >> choice;

do {

switch (choice) {

case 1: enter(); break; // Функция ввода

case 2: all\_out(); break;// Функция вывода с сортировкой

case 3: found(); break; // Функция поиска

}

} while (choice != 5);

}

void enter() {

int answer;

do {

if (now\_size >= size) { // проверка на место

cout << "Ошибка: массив поездов заполнен." << endl;

return;

}

cout << "Название санатория : ";

cin.ignore(); // Чистка буфера

getline(cin, health\_center[now\_size].name);

cout << "Местонахождение : ";

getline(cin, health\_center[now\_size].place);

cout << "Лечебный профиль: ";

cin.getline(health\_center[now\_size].healing,50);

cout << "Количество путевок: ";

cin >> health\_center[now\_size].travel\_packages;

now\_size++;

cout << "Продолжить ввод? (1 - да, 0 - нет): ";

cin >> answer;

} while (answer == 1);

cout << "Что дальше?" << endl;

cin >> choice;

}

void all\_out() {

//Сортировка по алфавиту...

for (int i = 0; i < now\_size - 1; i++) {

for (int j = 0; j < now\_size - i - 1; j++) {

if (strcmp(health\_center[j].healing, health\_center[j + 1].healing) > 0) {

swap(health\_center[j], health\_center[j + 1]);

}

}

}

for (int i = 0; i < now\_size; i++) { // Вывод

cout << "Название санатория : ";

cout << health\_center[i].name << endl;

cout << "Местонахождение : ";

cout << health\_center[now\_size].place << endl;

cout << "Лечебный профиль: ";

cout << health\_center[i].healing << endl;

cout << "Количество путевок: ";

cout << health\_center[i].travel\_packages << endl;

cout << endl;

}

cout << "Что дальше?" << endl;

cin >> choice;

}

void found() {

int ans, answer\_found;

string buffer; bool flag = false;

do{

cout << "Поиск по:" << endl;

cout << "1 - Название санатория : ";

cout << "2 - Местонахождение : ";

cout << "3 - Лечебный профиль: ";

cout << "4 - Количество путевок: ";

cin >> ans;

switch (ans) {

case 1:

cout << "Введите название санатория : ";

cin >> buffer;

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

if (buffer == health\_center[i].name) {

cout << "Название санатория : "<< health\_center[i].name << endl;

cout << "Местонахождение : "<< health\_center[i].place << endl;

cout << "Лечебный профиль: "<< health\_center[i].healing << endl;

cout << "Количество путевок: " << health\_center[i].travel\_packages << endl;

cout << endl;

}

}; break;

case 2:

cout << "Введите Местонахождение: ";

cin >> buffer;

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

if (buffer == health\_center[i].place) {

cout << "Название санатория : " << health\_center[i].name << endl;

cout << "Местонахождение : " << health\_center[i].place << endl;

cout << "Лечебный профиль: " << health\_center[i].healing << endl;

cout << "Количество путевок: " << health\_center[i].travel\_packages << endl;

cout << endl;

}

}; break;

case 3:

char buf[50];

cout << "Введите лечебный профиль : ";

cin.ignore(); // Чистка буфера

cin.getline(buf,50);

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

if (strcmp (buf, health\_center[i].healing) ==0) {

cout << "Название санатория : " << health\_center[i].name << endl;

cout << "Местонахождение : " << health\_center[i].place << endl;

cout << "Лечебный профиль: " << health\_center[i].healing << endl;

cout << "Количество путевок: " << health\_center[i].travel\_packages << endl;

cout << endl;

}

}; break;

case 4:

int reg;

cout << "Введите Количество путевок: ";

cin >> reg;

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

if (reg == health\_center[i].travel\_packages ) {

cout << "Название санатория : " << health\_center[i].name << endl;

cout << "Местонахождение : " << health\_center[i].place << endl;

cout << "Лечебный профиль: " << health\_center[i].healing << endl;

cout << "Количество путевок: " << health\_center[i].travel\_packages << endl;

cout << endl;

}

}; break;

}

cout << "Хотите продолжить поиск?(1 - да, 2 - нет)" << endl;

cin >> answer\_found;

} while (answer\_found == 1);

cout << "Что дальше?" << endl;

cin >> choice;

}

