

Отчет по лабораторной 4 2

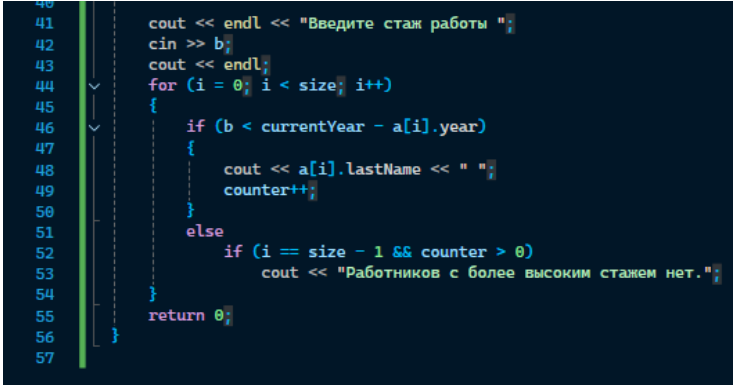
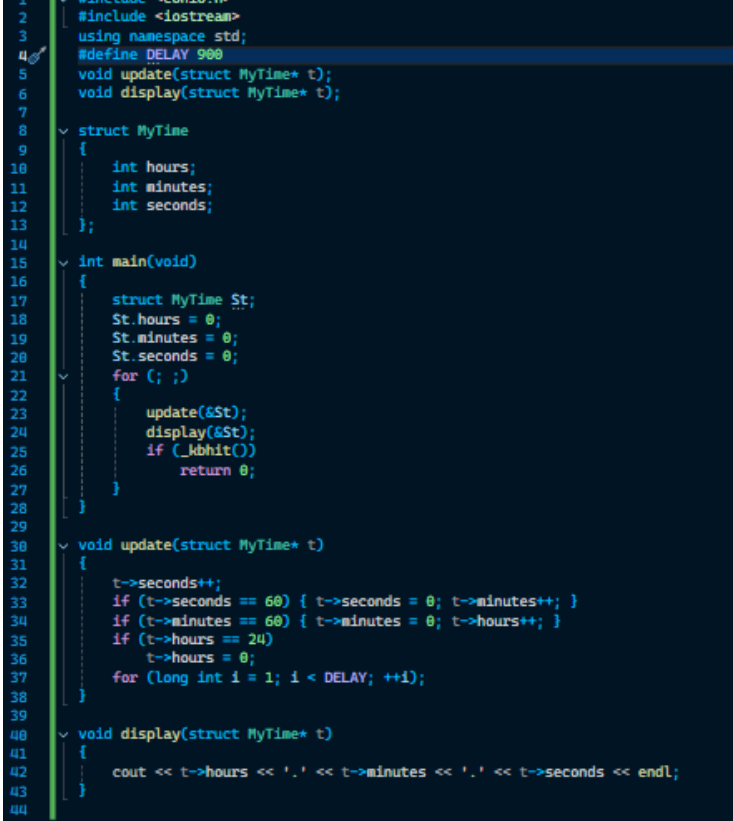
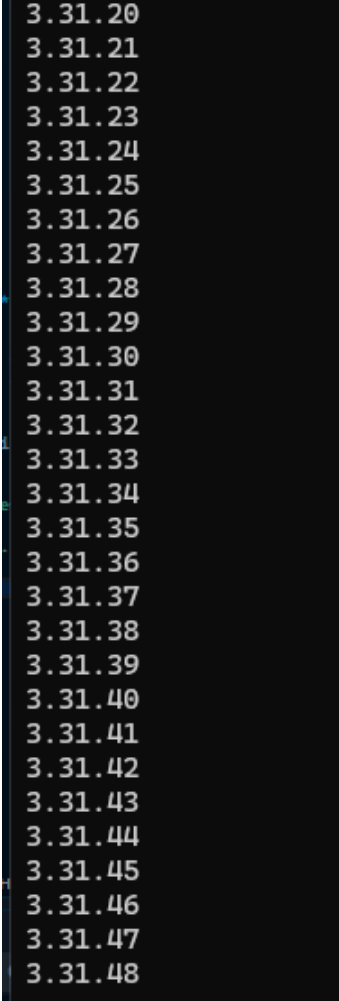
Задания

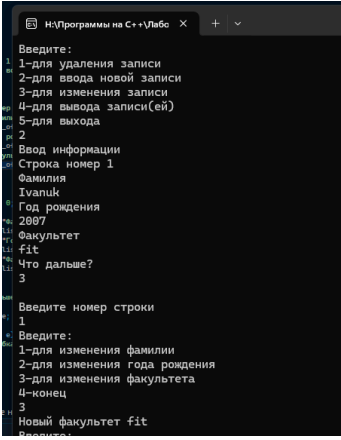
1	<p>1. Изучить способы организации данных в виде <i>структуры</i>, выполнив программу, записанную в данном пункте.</p> <p>Дополнить структуру дополнительными сведениями о работниках и вывести их вместе с фамилиями на экран.</p>
2	<p>2. В правой части приведен пример программы, в которой используется <i>указатель на структуру</i>.</p> <p>Выполнить программу, меняя скорость вывода информации.</p> <p>Написать пояснения к программе.</p>

1	<p>1. Изучить способы организации данных в виде <i>структуры</i>, выполнив программу, записанную в данном пункте.</p> <p>Дополнить структуру дополнительными сведениями о работниках и вывести их вместе с фамилиями на экран.</p>	
3	<p>3. В правой части приведен пример программы, которая осуществляет работу с данными, организованными в виде структуры, содержащей информацию о студентах: №, Фамилия, Год рождения, Факультет.</p> <p>Проанализировать работу главной функции и функций, входящих в программу.</p> <p>Добавить операторы выдачи сообщений об ошибках при вводе данных неправильного типа.</p>	
4	<p>4. Выполнить программу, приведенную в правой части, несколько раз с различными исходными данными.</p> <p>Убедиться, что данные, введенные с клавиатуры, записываются в файл с именем base.bin.</p> <p>Записать условие задачи и комментарии к программе.</p>	
5	<p>7</p> <p>Авиарейсы. Номер рейса, пункт назначения, время вылета, дата вылета, стоимость билета, количество мест. Выбор по пункту назначения.</p>	

1	<p>1. Изучить способы организации данных в виде <i>структуры</i>, выполнив программу, записанную в данном пункте.</p> <p>Дополнить структуру дополнительными сведениями о работниках и вывести их вместе с фамилиями на экран.</p>
Доп	<p>5. Дополнительные задания</p> <p>1. Определить структуру для представления информации о сданных студентом экзаменах, содержащую поля: ФИО студента, число экзаменов, полученные оценки. Определить функции для обработки отдельного объекта (например, для проверки, сданы ли все экзамены на 4 и 5). Написать функцию для обработки массива структур. В результате обработки требуется вычислить характеристику успеваемости студентов, то есть отношение числа студентов, сдавших экзамены на 4 и 5, к общему числу студентов, в процентах.</p> <p>2. Описать структуру с именем TRAIN, содержащую поля: названия пункта назначения, номер поезда, время отправления. Написать программу, выполняющую ввод с клавиатуры данных в массив, состоящий из восьми элементов типа TRAIN (записи должны быть размещены в алфавитном порядке по названиям пунктов назначения); вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времени (если таких поездов нет, то вывести сообщение об этом).</p> <p>3. Определить структуру для представления информации о наличии санаторных путевок, содержащую следующие поля: Название санатория, Место расположения, Лечебный профиль, Количество путевок. Представить введенные данные в виде таблицы, сгруппировав их по лечебным профилям санаториев. В пределах каждой группы данные отсортировать по названиям санаториев. Организовать поиск информации и вывод результатов.</p>


Номер задания	Скриншот кода	Скриншот результата
1	<pre> 1 #include <iostream> 2 using namespace std; 3 4 struct Worker 5 { 6 char lastName[30]; 7 char name[30]; 8 char position[30]; 9 int year; 10 int age; 11 } 12 13 14 int main() 15 { 16 setlocale(LC_ALL, "Russian"); 17 const int size = 3; 18 const int currentYear = 2025; 19 int i, b, counter = 0; 20 Worker a[size]; 21 for (i = 0; i < size; i++) 22 { 23 cout << "Введите фамилию " << i + 1 << "-ого работника " << endl; 24 cin >> a[i].lastName; 25 cout << "Введите имя " << i + 1 << "-ого работника " << endl; 26 cin >> a[i].name; 27 cout << endl << "Должность: "; 28 cin >> a[i].position; 29 cout << endl << "Год поступления на работу: "; 30 cin >> a[i].year; 31 cout << endl << "Возраст: "; 32 cin >> a[i].age; 33 } 34 cout << endl << "Список сотрудников: "; 35 36 for (int i = 0; i < size; i++) { 37 cout << endl << a[i].lastName << " " << a[i].name << 38 " " << a[i].position << " " << a[i].age << " y.o" 39 } </pre>	<p>Год поступления на работу: 2020</p> <p>Возраст: 17 Введите фамилию 2-ого работника Lavshuk Введите имя 2-ого работника Stasik</p> <p>Должность: intern</p> <p>Год поступления на работу: 2025</p> <p>Возраст: 18 Введите фамилию 3-ого работника Glezov Введите имя 3-ого работника Vitalya</p> <p>Должность: junior</p> <p>Год поступления на работу: 2023</p> <p>Возраст: 18</p> <p>Список сотрудников: Ivanuk Alexei Lead 17 y.o Lavshuk Stasik intern 18 y.o Glezov Vitalya junior 18 y.o Введите стаж работы 1</p> <p>Ivanuk Glezov</p>

Номер задания	Скриншот кода	Скриншот результата
	 <pre> 40 41 cout << endl << "Введите стаж работы " << b; 42 cin >> b; 43 cout << endl; 44 for (i = 0; i < size; i++) 45 { 46 if (b < currentYear - a[i].year) 47 { 48 cout << a[i].lastName << " "; 49 counter++; 50 } 51 else 52 if (i == size - 1 && counter > 0) 53 cout << "Работников с более высоким стажем нет." << endl; 54 } 55 return 0; 56 } 57 </pre>	
2	 <pre> 1 #include <conio.h> 2 #include <iostream> 3 using namespace std; 4 #define DELAY 900 5 void update(struct MyTime* t); 6 void display(struct MyTime* t); 7 8 struct MyTime 9 { 10 int hours; 11 int minutes; 12 int seconds; 13 }; 14 15 int main(void) 16 { 17 struct MyTime St; 18 St.hours = 0; 19 St.minutes = 0; 20 St.seconds = 0; 21 for (; ;) 22 { 23 update(&St); 24 display(&St); 25 if (_kbhit()) 26 return 0; 27 } 28 } 29 30 void update(struct MyTime* t) 31 { 32 t->seconds++; 33 if (t->seconds == 60) { t->seconds = 0; t->minutes++; } 34 if (t->minutes == 60) { t->minutes = 0; t->hours++; } 35 if (t->hours == 24) 36 t->hours = 0; 37 for (long int i = 1; i < DELAY; ++i); 38 } 39 40 void display(struct MyTime* t) 41 { 42 cout << t->hours << '.' << t->minutes << '.' << t->seconds << endl; 43 } 44 </pre>	 <pre> 3.31.20 3.31.21 3.31.22 3.31.23 3.31.24 3.31.25 3.31.26 3.31.27 3.31.28 3.31.29 3.31.30 3.31.31 3.31.32 3.31.33 3.31.34 3.31.35 3.31.36 3.31.37 3.31.38 3.31.39 3.31.40 3.31.41 3.31.42 3.31.43 3.31.44 3.31.45 3.31.46 3.31.47 3.31.48 </pre>
	<p>Программа реализует простой таймер, который отслеживает и обновляет время в формате "часы.минуты.секунды" с использованием структуры MyTime .</p>	

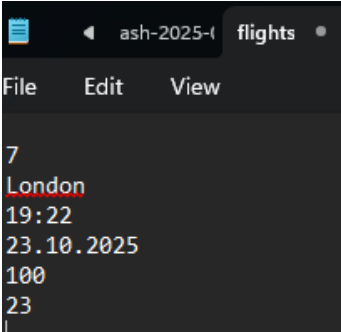
Номер задания	Скриншот кода	Скриншот результата
3	<pre> 1 #include <iostream> 2 # define str_len 30 3 # define size 30 4 using namespace std; 5 void enter_new(); 6 void del(); 7 void change(); 8 void out(); 9 struct Student 10 { 11 char name[str_len]; 12 int year_of_birth; 13 char department[5]; 14 }; 15 struct Student list_of_student[size]; 16 struct Student bad; 17 int current_size = 0; int choice; 18 int main() 19 { 20 setlocale(LC_CTYPE, "Russian"); 21 cout << "Введите:" << endl; 22 cout << "1-для удаления записи" << endl; 23 cout << "2-для ввода новой записи" << endl; 24 cout << "3-для изменения записи" << endl; 25 cout << "4-для вывода записи(ей)" << endl; 26 cout << "5-для выхода" << endl; 27 try { 28 cin >> choice; 29 } 30 catch (exception e) { 31 cout << "Ошибка выбора!"; 32 choice = 5; 33 } 34 do 35 { 36 switch (choice) 37 { 38 case 1: del(); break; 39 case 2: enter_new(); break; 40 case 3: change(); break; 41 case 4: out(); break; 42 } 43 while (choice != 5); </pre>	
	<pre> 44 } 45 void enter_new() 46 { 47 cout << "Ввод информации" << endl; 48 if (current_size < size) 49 { 50 cout << "Строка номер "; 51 cout << current_size + 1; 52 cout << endl << "Фамилия " << endl; 53 cin >> list_of_student[current_size].name; 54 cout << "Год рождения " << endl; 55 cin >> list_of_student[current_size].year_of_birth; 56 cout << "Факультет " << endl; 57 cin >> list_of_student[current_size].department; 58 current_size++; 59 } 60 else 61 cout << "Введено максимальное кол-во строк"; 62 cout << "Что дальше?" << endl; 63 try { 64 cin >> choice; 65 } 66 catch (exception e) { 67 cout << "Ошибка выбора!"; 68 choice = 5; 69 } 70 } 71 void del() 72 { 73 int d; 74 cout << "Номер строки, которую надо удалить (для удаления всех строк нажать 99)" << endl; cin >> d; 75 if (d != 99) 76 { 77 for (int del = (d - 1); del < current_size; del++) 78 list_of_student[del] = list_of_student[del + 1]; 79 current_size = current_size - 1; 80 } 81 if (d == 99) 82 for (int i = 0; i < size; i++) 83 list_of_student[i] = bad; 84 cout << "Что дальше?" << endl; </pre>	

Номер задания	Скриншот кода	Скриншот результата
	<pre> 85 try { 86 cin >> choice; 87 } 88 catch (exception e) { 89 cout << "Ошибка выбора!"; 90 choice = 5; 91 } 92 } 93 void change() 94 { 95 int n, per; 96 cout << "\nВведите номер строки" << endl; cin >> n; 97 do 98 { 99 cout << "Введите: " << endl; 100 cout << "1-для изменения фамилии" << endl; 101 cout << "2-для изменения года рождения" << endl; 102 cout << "3-для изменения факультета" << endl; 103 cout << "4-конец\n"; 104 try { 105 cin >> per; 106 } 107 catch (exception e) { 108 cout << "Ошибка выбора!"; 109 choice = 5; 110 } 111 switch (per) 112 { 113 case 1: cout << "Новая фамилия"; 114 cin >> list_of_student[n - 1].name; break; 115 case 2: cout << "Новый год рождения"; 116 cin >> list_of_student[n - 1].year_of_birth; break; 117 case 3: cout << "Новый факультет "; 118 cin >> list_of_student[n - 1].department; break; 119 } 120 } while (per != 4); 121 cout << "Что дальше?" << endl; 122 try { </pre>	
	<pre> 122 cout << "Что дальше?" << endl; 123 try { 124 cin >> choice; 125 } 126 catch (exception e) { 127 cout << "Ошибка выбора!"; 128 choice = 5; 129 } 130 void out() 131 { 132 int sw, n; 133 cout << "1-вывод 1 строки" << endl; 134 cout << "2-вывод всех строк" << endl; 135 cin >> sw; 136 if (sw == 1) 137 { 138 cout << "Номер выводимой строки " << endl; cin >> n; cout << endl; 139 cout << "Фамилия "; 140 cout << list_of_student[n - 1].name << endl; 141 cout << "Год рождения "; 142 cout << list_of_student[n - 1].year_of_birth << endl; 143 cout << "Факультет "; 144 cout << list_of_student[n - 1].department << endl; 145 } 146 if (sw == 2) 147 { 148 for (int i = 0; i < current_size; i++) 149 { 150 cout << "Фамилия "; 151 cout << list_of_student[i].name << endl; 152 cout << "Год рождения "; 153 cout << list_of_student[i].year_of_birth << endl; 154 cout << "Факультет "; 155 cout << list_of_student[i].department << endl; 156 } 157 } 158 cout << "Что дальше?" << endl; 159 try { 160 cin >> choice; 161 } 162 catch (exception e) { 163 cout << "Ошибка выбора!"; 164 choice = 5; 165 } 166 } 167 </pre>	

Номер задания	Скриншот кода	Скриншот результата
4	<pre>4 void output(); // Вывод данных 5 void find(char lastName[]); // Поиск данных 6 7 // Структура для хранения информации о студентах 8 typedef struct Students 9 { 10 char fio[16]; 11 char group[3]; 12 } STUD; 13 14 int number; // Количество студентов 15 FILE* f; // Указатель на файл 16 errno_t err; // Код ошибки 17 18 int main() 19 { 20 setlocale(LC_ALL, "Russian"); 21 int choice; // Выбор действия 22 char fio[16]; 23 24 // Главное меню 25 do 26 { 27 cout << "\n1.Ввод данных с клавиатуры и запись в файл\n"; 28 cout << "2.Вывод данных из файла\n"; 29 cout << "3.Поиск по фамилии\n"; 30 cout << "0.Выход из программы\n\n"; 31 cout << "Введите номер операции: "; 32 cin >> choice; 33 34 // Обработка выбора пользователя 35 switch (choice) 36 { 37 case 1: // Ввод данных 38 cout << "Введите количество студентов: "; 39 cin >> number; 40 input(number); 41 break; 42 43 case 2: // Вывод данных из файла 44 output(); 45 break; 46 47 case 3: // Поиск по фамилии 48 cout << "Введите фамилию: "; 49 cin >> fio; 50 find(fio); 51 break; 52 53 case 0: // Выход из программы 54 exit(0); 55 break; 56 } 57 } while (choice != 0); // Цикл выполняется, пока не выбрано 0 58 59 // Функция ввода данных</pre>	<p>1.Ввод данных с клавиатуры и запись в файл 2.Вывод данных из файла 3.Поиск по фамилии 0.Выход из программы</p> <p>Введите номер операции: 1 Введите количество студентов: 1 Фамилия: Makarevich Группа: 10</p> <p>1.Ввод данных с клавиатуры и запись в файл 2.Вывод данных из файла 3.Поиск по фамилии 0.Выход из программы</p> <p>Введите номер операции: 3 Введите фамилию: Shiman</p> <p>Фамилия Группа Shiman 1</p> <p>1.Ввод данных с клавиатуры и запись в файл 2.Вывод данных из файла 3.Поиск по фамилии 0.Выход из программы</p> <p>Введите номер операции: 2</p> <p>Фамилия Группа Shiman 1 Beloded 2 Snelov 3 Makarevich 10</p>
	<pre>61 void input(int size) 62 { 63 STUD buf = { ' ', ' ', ' ' }; // Буфер для записи данных 64 if (!fopen_s(&f, "base.bin", "ab")) // Открытие файла для добавления 65 { 66 for (int p = 0; p < size; p++) 67 { 68 cout << "Фамилия: "; 69 cin >> buf.fio; // Ввод фамилии 70 cout << "Группа: "; 71 cin >> buf.group; // Ввод группы 72 fwrite(&buf, sizeof(buf), 1, f); // Запись данных в файл 73 } 74 fclose(f); // Закрытие файла 75 } 76 else { 77 cout << "Ошибка открытия файла"; 78 return; 79 } 80 } 81 82 // Функция вывода данных из файла 83 void output() 84 { 85 STUD buf; // Буфер для чтения данных 86 if (!fopen_s(&f, "base.bin", "rb")) // Открытие файла для чтения 87 { 88 cout << "\nФамилия Группа\n"; 89 fread(&buf, sizeof(buf), 1, f); // Чтение первой записи 90 while (!feof(f)) // Пока не конец файла 91 { 92 cout << buf.fio << "\t" << buf.group << endl; // Вывод данных 93 fread(&buf, sizeof(buf), 1, f); // Чтение следующей записи 94 } 95 cout << endl; 96 fclose(f); // Закрытие файла 97 } 98 else { 99 cout << "Ошибка открытия файла"; // Обработка ошибки 100 return; 101 } 102 } 103 104 }</pre>	

Номер задания	Скриншот кода	Скриншот результата
	 <pre> 105 // Функция поиска по фамилии 106 void find(char lastName[16]) 107 { 108 bool flag = false; // Флаг для проверки нахождения записи 109 STUD buf; // Буфер для чтения данных 110 if (!fopen_s(&f, "base.bin", "rb")) // Открытие файла для чтения 111 { 112 while (!feof(f)) // Пока не конец файла 113 { 114 fread(&buf, sizeof(buf), 1, f); // Чтение записи 115 if (strcmp(lastName, buf.fio) == 0) // Сравнение фамилии 116 { 117 cout << "\nФамилия Группа\n"; 118 cout << buf.fio << "\t " << buf.group << endl; // Вывод найденной записи 119 flag = true; 120 break; 121 } 122 } 123 fclose(f); // Закрытие файла 124 if (!flag) 125 cout << "Ничего не найдено\n"; 126 } 127 else 128 { 129 cout << "Ошибка открытия файла"; 130 return; 131 } 132 } 133 </pre>	
	<p>Программа позволяет пользователю вводить данные студентов (фамилию и номер группы) с клавиатуры и записывать их в файл, выводить на экран содержимое файла, а также искать информацию о студенте по введенной фамилии.</p>	

Номер задания	Скриншот кода	Скриншот результата
5	<pre>1 #include <iostream> 2 #include <fstream> 3 #include <string> 4 using namespace std; 5 6 struct AVIALINES { 7 int flightNumber; // Номер рейса 8 string destination; // Пункт назначения 9 string departureTime; // Время вылета 10 string departureDate; // Дата вылета 11 double ticketPrice; // Стоимость билета 12 int seatCount; // Количество мест 13 }; 14 15 const int MAX_LINES = 100; 16 AVIALINES lines[MAX_LINES]; 17 int lineCount = 0; 18 19 void addLine(); 20 void displayLines(); 21 void deleteLine(); 22 void searchLine(); 23 void saveToFile(); 24 void loadFromFile(); 25 26 int main() { 27 int choice; 28 setlocale(LC_ALL, "rus"); 29 do { 30 cout << "1 - Добавить рейс\n"; 31 cout << "2 - Показать все рейсы\n"; 32 cout << "3 - Удалить рейс\n"; 33 cout << "4 - Поиск рейса по пункту назначения\n"; 34 cout << "5 - Сохранить в файл\n"; 35 cout << "6 - Загрузить из файла\n"; 36 cout << "7 - Выход\n"; 37 cout << "Выберите действие: "; 38 cin >> choice; 39 40 switch (choice) { 41 case 1: 42 addLine(); 43 break; 44 case 2: 45 displayLines(); 46 break; 47 case 3: 48 deleteLine(); 49 break; 50 case 4: 51 searchLine(); 52 break;</pre>	<pre>1 - Добавить рейс 2 - Показать все рейсы 3 - Удалить рейс 4 - Поиск рейса по пункту назначения 5 - Сохранить в файл 6 - Загрузить из файла 7 - Выход Выберите действие: 1 Введите номер рейса: 7 Введите пункт назначения: London Введите время вылета (чч:мм): 19:00 Введите дату вылета (дд.мм.гггг): 20.10.202 Введите стоимость билета: 120 Введите количество мест: 80 Рейс добавлен! 1 - Добавить рейс 2 - Показать все рейсы 3 - Удалить рейс 4 - Поиск рейса по пункту назначения 5 - Сохранить в файл 6 - Загрузить из файла 7 - Выход Выберите действие: 5 Данные успешно сохранены в файл. 1 - Добавить рейс 2 - Показать все рейсы 3 - Удалить рейс 4 - Поиск рейса по пункту назначения 5 - Сохранить в файл 6 - Загрузить из файла 7 - Выход Выберите действие: 2 Рейс #1: Номер рейса: 7 Пункт назначения: London Время вылета: 19:00 Дата вылета: 20.10.2025 Стоимость билета: 120 Количество мест: 80 1 - Добавить рейс 2 - Показать все рейсы 3 - Удалить рейс 4 - Поиск рейса по пункту назначения 5 - Сохранить в файл 6 - Загрузить из файла 7 - Выход Выберите действие: 7 Выход из программы.</pre>

Номер задания	Скриншот кода	Скриншот результата
	<pre> 52 break; 53 case 5: 54 saveToFile(); 55 break; 56 case 6: 57 loadFromFile(); 58 break; 59 case 7: 60 cout << "Выход из программы.\n"; 61 break; 62 default: 63 cout << "Неверный выбор!\n"; 64 } 65 } while (choice != 0); 66 67 return 0; 68 } 69 70 void addLine() { 71 if (lineCount >= MAX_LINES) { 72 cout << "Массив заполнен. Невозможно добавить больше рейсов.\n"; 73 return; 74 } 75 AVIALINES line; 76 cout << "Введите номер рейса: "; 77 cin >> line.flightNumber; 78 cin.ignore(); 79 cout << "Введите пункт назначения: "; 80 getline(cin, line.destination); 81 cout << "Введите время вылета (чч:мм): "; 82 getline(cin, line.departureTime); 83 cout << "Введите дату вылета (дд.мм.rrrr): "; 84 getline(cin, line.departureDate); 85 cout << "Введите стоимость билета: "; 86 cin >> line.ticketPrice; 87 cout << "Введите количество мест: "; 88 cin >> line.seatCount; 89 90 lines[lineCount++] = line; 91 cout << "Рейс добавлен!\n"; 92 } 93 94 void displayLines() { 95 if (lineCount == 0) { 96 cout << "Список рейсов пуст.\n"; 97 return; 98 } </pre>	 <pre> ash-2025- flights File Edit View 7 London 19:22 23.10.2025 100 23 </pre>
	<pre> 99 for (int i = 0; i < lineCount; i++) { 100 cout << "\nРейс #" << i + 1 << ":\n"; 101 cout << "Номер рейса: " << lines[i].flightNumber << "\n"; 102 cout << "Пункт назначения: " << lines[i].destination << "\n"; 103 cout << "Время вылета: " << lines[i].departureTime << "\n"; 104 cout << "Дата вылета: " << lines[i].departureDate << "\n"; 105 cout << "Стоимость билета: " << lines[i].ticketPrice << "\n"; 106 cout << "Количество мест: " << lines[i].seatCount << "\n"; 107 } 108 } 109 110 void deleteLine() { 111 int flightNumber; 112 cout << "Введите номер рейса для удаления: "; 113 cin >> flightNumber; 114 int index = -1; 115 for (int i = 0; i < lineCount; i++) { 116 if (lines[i].flightNumber == flightNumber) { 117 index = i; 118 break; 119 } 120 } 121 122 if (index == -1) { 123 cout << "Рейс с таким номером не найден :/\n"; 124 return; 125 } 126 for (int i = index; i < lineCount - 1; i++) { 127 lines[i] = lines[i + 1]; 128 } 129 --lineCount; 130 cout << "Рейс успешно удален.\n"; 131 } 132 133 void searchLine() { 134 string destination; 135 cout << "Введите пункт назначения для поиска: "; 136 cin.ignore(); 137 getline(cin, destination); 138 139 bool isFound = false; 140 for (int i = 0; i < lineCount; i++) { 141 if (lines[i].destination == destination) { 142 cout << "\nНайден рейс:\n"; 143 cout << "Номер рейса: " << lines[i].flightNumber << "\n"; 144 cout << "Время вылета: " << lines[i].departureTime << "\n"; 145 cout << "Дата вылета: " << lines[i].departureDate << "\n"; 146 cout << "Стоимость билета: " << lines[i].ticketPrice << "\n"; 147 cout << "Количество мест: " << lines[i].seatCount << "\n"; 148 isFound = true; </pre>	

Номер задания	Скриншот кода	Скриншот результата
	<pre> 148 } 149 } 150 } 151 152 if (!isFound) { 153 cout << "Рейсы с таким пунктом назначения не найдены.\n"; 154 } 155 } 156 157 void saveToFile() { 158 ofstream file("avialines.txt"); 159 if (!file) { 160 cout << "Ошибка при открытии файла.\n"; 161 return; 162 } 163 file << lineCount << "\n"; 164 for (int i = 0; i < lineCount; i++) { 165 file << lines[i].flightNumber << "\n" 166 << lines[i].destination << "\n" 167 << lines[i].departureTime << "\n" 168 << lines[i].departureDate << "\n" 169 << lines[i].ticketPrice << "\n" 170 << lines[i].seatCount << "\n"; 171 } 172 file.close(); 173 cout << "Данные успешно сохранены в файл.\n"; 174 } 175 176 void loadFromFile() { 177 ifstream file("lines.txt"); 178 if (!file) { 179 cout << "Ошибка при открытии файла.\n"; 180 return; 181 } 182 file >> lineCount; 183 file.ignore(); 184 for (int i = 0; i < lineCount; i++) { 185 file >> lines[i].flightNumber; 186 file.ignore(); 187 getline(file, lines[i].destination); 188 getline(file, lines[i].departureTime); 189 getline(file, lines[i].departureDate); 190 file >> lines[i].ticketPrice; 191 file >> lines[i].seatCount; 192 file.ignore(); 193 } 194 file.close(); 195 cout << "Данные успешно загружены из файла.\n"; 196 } 197 </pre>	
Доп 1	<pre> 1 #include <iostream> 2 #include <string> 3 4 using namespace std; 5 6 struct Student { 7 string fullName; // ФИО студента 8 int examCount; // Число экзаменов 9 int grades[10]; // оценки 10 }; 11 12 bool isGreat(const Student& student) { 13 for (int i = 0; i < student.examsCount; i++) { 14 if (student.grades[i] < 4) { 15 return false; 16 } 17 } 18 return true; 19 } 20 21 double calculateActivity(Student students[], int studentCount) { 22 int greatStudents = 0; 23 for (int i = 0; i < studentCount; i++) { 24 if (isGreat(students[i])) { 25 greatStudents++; 26 } 27 } 28 return (studentCount == 0) ? 0.0 : (static_cast<double>(greatStudents) / studentCount) * 100; 29 } 30 31 void main() { 32 setlocale(LC_ALL, "rus"); 33 int studentCount; 34 cout << "Введите количество студентов: "; 35 cin >> studentCount; 36 if (studentCount <= 0 studentCount > 100) { 37 cout << "Некорректное количество студентов!" << endl; 38 return; 39 } </pre>	<pre> Введите количество студентов: 3 Студент #1: Введите ФИО: Ivanuk Введите количество экзаменов: 4 Введите оценки (через пробел): 4 4 5 10 Студент #2: Введите ФИО: Lavshuk Stasik Введите количество экзаменов: 4 Введите оценки (через пробел): 6 9 6 9 Студент #3: Введите ФИО: Makarevich Kirill Введите количество экзаменов: 4 Введите оценки (через пробел): 4 4 5 9 Информация о студентах: ФИО: Ivanuk, Экзаменов: 4, Оценки: 4 4 5 10 (Все на 4 и 5) ФИО: Lavshuk Stasik, Экзаменов: 4, Оценки: 6 9 6 9 (Все на 4 и 5) ФИО: Makarevich Kirill, Экзаменов: 4, Оценки: 4 4 5 9 (Все на 4 и 5) Характеристика успеваемости: 100% </pre>

Номер задания	Скриншот кода	Скриншот результата
	<pre> 39 40 41 Student students[100]; 42 for (int i = 0; i < studentCount; i++) { 43 44 cout << "\nСтудент #" << (i + 1) << ":\n"; 45 cout << "Введите ФИО: "; 46 cin.ignore(); 47 getline(cin, students[i].fullName); 48 49 cout << "Введите количество экзаменов: "; 50 cin >> students[i].examCount; 51 52 if (students[i].examCount <= 0 students[i].examCount > 10) { 53 cout << "Некорректное количество экзаменов!" << endl; 54 return; 55 } 56 57 cout << "Введите оценки (через пробел): "; 58 for (int j = 0; j < students[i].examCount; j++) { 59 cin >> students[i].grades[j]; 60 } 61 62 cout << "\nИнформация о студентах:\n"; 63 for (int i = 0; i < studentCount; i++) { 64 cout << "ФИО: " << students[i].fullName 65 << ", Экзаменов: " << students[i].examCount 66 << ", Оценки: "; 67 for (int j = 0; j < students[i].examCount; j++) { 68 cout << students[i].grades[j] << " "; 69 } 70 cout << (isGreat(students[i]) ? "(Все на 4 и 5)" : "(Есть оценки ниже 4)") << endl; 71 } 72 double performance = calculateActivity(students, studentCount); 73 cout << "\nХарактеристика успеваемости: " << performance << "%" << endl; 74 } 75 </pre>	
Доп 2	<pre> 1 #include <iostream> 2 #include <string> 3 4 using namespace std; 5 6 struct TRAIN { 7 string destination; // Название пункта назначения 8 int trainNumber; // Номер поезда 9 string departureTime; // Время 10 }; 11 12 bool isAfterTime(const string& trainTime, const string& inputTime) { 13 return trainTime > inputTime; 14 } 15 16 void sortTrainsByDestination(TRAIN trains[], int count) { 17 for (int i = 0; i < count - 1; i++) { 18 for (int j = 0; j < count - i - 1; j++) { 19 if (trains[j].destination > trains[j + 1].destination) { 20 TRAIN tmp = trains[j]; 21 trains[j] = trains[j + 1]; 22 trains[j + 1] = tmp; 23 } 24 } 25 } 26 } 27 28 void main() { 29 setlocale(LC_ALL, "rus"); 30 const int trainCount = 8; 31 TRAIN trains[trainCount]; 32 cout << "Введите данные о " << trainCount << " поездах:\n"; 33 for (int i = 0; i < trainCount; i++) { 34 cout << "\nПоезд #" << (i + 1) << ":\n"; 35 36 cout << "Введите пункт назначения: "; 37 cin.ignore(); 38 getline(cin, trains[i].destination); 39 </pre>	<p>Введите данные о 8 поездах:</p> <p>Поезд #1: Введите пункт назначения: Minsk Введите номер поезда: 7 Введите время отправления: 12:32</p> <p>Поезд #2: Введите пункт назначения: Gomel Введите номер поезда: 2 Введите время отправления: 4:56</p> <p>Поезд #3: Введите пункт назначения: Rechitsa Введите номер поезда: 3 Введите время отправления: 20:45</p> <p>Поезд #4: Введите пункт назначения: Pinsk Введите номер поезда: 9 Введите время отправления: 19:34</p> <p>Поезд #5: Введите пункт назначения: Volozhin Введите номер поезда: 12 Введите время отправления: 18:09</p> <p>Поезд #6: Введите пункт назначения: Orsha Введите номер поезда: 5 Введите время отправления: 6:18</p> <p>Поезд #7: Введите пункт назначения: Mogilev Введите номер поезда: 5 Введите время отправления: 10:27</p> <p>Поезд #8: Введите пункт назначения: Bobruisk Введите номер поезда: 8 Введите время отправления: 15:02</p> <p>Введите время, чтобы вывести поезда, отправляющиеся после него: 12:00</p> <p>Поезда, отправляющиеся после 12:00: Пункт назначения: Bobruisk, Номер поезда: 8, Время отправления: 15:02 Пункт назначения: Gomel, Номер поезда: 2, Время отправления: 4:56 Пункт назначения: Orsha, Номер поезда: 5, Время отправления: 6:18 Пункт назначения: Pinsk, Номер поезда: 9, Время отправления: 19:34 Пункт назначения: Rechitsa, Номер поезда: 3, Время отправления: 20:45 Пункт назначения: Volozhin, Номер поезда: 12, Время отправления: 18:09 Пункт назначения: Mogilev, Номер поезда: 7, Время отправления: 12:32</p>
	<pre> 40 cout << "Введите номер поезда: "; 41 cin >> trains[i].trainNumber; 42 43 cout << "Введите время отправления: "; 44 cin >> trains[i].departureTime; 45 } 46 47 sortTrainsByDestination(trains, trainCount); 48 49 string inputTime; 50 cout << "\nВведите время, чтобы вывести поезда, отправляющиеся после него: "; 51 cin >> inputTime; 52 53 cout << "\nПоезда, отправляющиеся после " << inputTime << ":\n"; 54 bool found = false; 55 for (int i = 0; i < trainCount; i++) { 56 if (isAfterTime(trains[i].departureTime, inputTime)) { 57 cout << "Пункт назначения: " << trains[i].destination 58 << ", Номер поезда: " << trains[i].trainNumber 59 << ", Время отправления: " << trains[i].departureTime << endl; 60 found = true; 61 } 62 } 63 if (!found) { 64 cout << "Нет поездов, отправляющихся после " << inputTime << ":\n"; 65 } 66 } 67 </pre>	

Номер задания	Скриншот кода	Скриншот результата																														
Доп 3	<pre>1 #include <iostream> 2 #include <string> 3 #include <windows.h> 4 using namespace std; 5 6 struct Sanatorium { 7 string name; // Название санатория 8 string location; // расположение 9 string profile; // лечебный профиль 10 int voucherCount; // путевки 11 }; 12 13 void sortSanatoriums(Sanatorium sanatoriums[], int count) { 14 for (int i = 0; i < count - 1; i++) { 15 for (int j = 0; j < count - i - 1; j++) { 16 if (sanatoriums[j].profile > sanatoriums[j + 1].profile 17 (sanatoriums[j].profile == sanatoriums[j + 1].profile && sanatoriums[j].name > sanatoriums[j + 1].name)) { 18 Sanatorium tmp = sanatoriums[j]; 19 sanatoriums[j] = sanatoriums[j + 1]; 20 sanatoriums[j + 1] = tmp; 21 } 22 } 23 } 24 } 25 26 void printSanatoriums(Sanatorium sanatoriums[], int count) { 27 cout << "\nСанатории:\n"; 28 cout << "-----\n"; 29 cout << " Лечебный профиль Название санатория Местоположение Путевки \n"; 30 cout << "-----\n"; 31 32 for (int i = 0; i < count; i++) { 33 cout << " " << sanatoriums[i].profile 34 << string(19 - sanatoriums[i].profile.length(), ' ') << " " 35 << sanatoriums[i].name 36 << string(25 - sanatoriums[i].name.length(), ' ') << " " 37 << sanatoriums[i].location 38 << string(20 - sanatoriums[i].location.length(), ' ') << " " 39 << sanatoriums[i].voucherCount << " \n"; 40 } 41 42 cout << "-----\n"; 43 }</pre>	<p>Санаторий #1: Введите название санатория: Березина Введите место расположения: Минская область Введите лечебный профиль: Кардиология Введите количество путевок: 80</p> <p>Санаторий #2: Введите название санатория: Янчукина Введите место расположения: Минская область Введите лечебный профиль: Теряния Введите количество путевок: 120</p> <p>Санаторий #3: Введите название санатория: Солнечное Введите место расположения: Минская область Введите лечебный профиль: Оттология Введите количество путевок: 60</p> <p>Санаторий #4: Введите название санатория: БГТУ Введите место расположения: Минск Введите лечебный профиль: Дуа Введите количество путевок: 150</p> <p>Санаторий #5: Введите название санатория: Случь Введите место расположения: Минская область Введите лечебный профиль: Кардиология Введите количество путевок: 100</p> <p>Санатории:</p> <table><thead><tr><th>Лечебный профиль</th><th>Название санатория</th><th>Местоположение</th><th>Путевки</th></tr></thead><tbody><tr><td>Кардиология</td><td>Случь</td><td>Минская область</td><td>100</td></tr><tr><td>Кардиология</td><td>Березина</td><td>Минская область</td><td>80</td></tr><tr><td>Оттология</td><td>Солнечное</td><td>Минская область</td><td>60</td></tr><tr><td>Теряния</td><td>Янчукина</td><td>Минская область</td><td>120</td></tr><tr><td>Дуа</td><td>БГТУ</td><td>Минск</td><td>150</td></tr></tbody></table> <p>Введите лечебный профиль для поиска: Кардиология</p> <p>Результаты поиска по лечебному профилю "Кардиология":</p> <table><thead><tr><th>Название санатория</th><th>Местоположение</th><th>Путевки</th></tr></thead><tbody><tr><td>Случь</td><td>Минская область</td><td>100</td></tr></tbody></table>	Лечебный профиль	Название санатория	Местоположение	Путевки	Кардиология	Случь	Минская область	100	Кардиология	Березина	Минская область	80	Оттология	Солнечное	Минская область	60	Теряния	Янчукина	Минская область	120	Дуа	БГТУ	Минск	150	Название санатория	Местоположение	Путевки	Случь	Минская область	100
Лечебный профиль	Название санатория	Местоположение	Путевки																													
Кардиология	Случь	Минская область	100																													
Кардиология	Березина	Минская область	80																													
Оттология	Солнечное	Минская область	60																													
Теряния	Янчукина	Минская область	120																													
Дуа	БГТУ	Минск	150																													
Название санатория	Местоположение	Путевки																														
Случь	Минская область	100																														
	<pre>46 void searchSanatoriums(Sanatorium sanatoriums[], int count, const string& searchProfile) { 47 cout << "\nРезультаты поиска по лечебному профилю \"" << searchProfile << ":\n"; 48 bool found = false; 49 50 for (int i = 0; i < count; i++) { 51 if (sanatoriums[i].profile == searchProfile) { 52 if (!found) { 53 cout << "-----\n"; 54 cout << " Название санатория Местоположение Путевки \n"; 55 cout << "-----\n"; 56 57 cout << " " << sanatoriums[i].name 58 << string(25 - sanatoriums[i].name.length(), ' ') << " " 59 << sanatoriums[i].location 60 << string(20 - sanatoriums[i].location.length(), ' ') << " " 61 << sanatoriums[i].voucherCount << " \n"; 62 63 found = true; 64 } 65 66 if (!found) { 67 cout << "Санаториев с таким лечебным профилем не найдено.\n"; 68 } 69 } 70 else { 71 cout << "-----\n"; 72 } 73 } 74 }</pre>																															
	<pre>74 void main() { 75 SetConsoleOutputCP(1251); 76 SetConsoleCP(1251); 77 const int sanatoriumCount = 5; 78 Sanatorium sanatoriums[sanatoriumCount]; 79 cout << "Введите данные о " << sanatoriumCount << " санаториях:\n"; 80 for (int i = 0; i < sanatoriumCount; i++) { 81 cout << "\nСанаторий #" << (i + 1) << ":\n"; 82 cout << "Введите название санатория: "; 83 cin.ignore(); 84 getline(cin, sanatoriums[i].name); 85 cout << "Введите место расположения: "; 86 getline(cin, sanatoriums[i].location); 87 cout << "Введите лечебный профиль: "; 88 getline(cin, sanatoriums[i].profile); 89 cout << "Введите количество путевок: "; 90 cin >> sanatoriums[i].voucherCount; 91 } 92 sortSanatoriums(sanatoriums, sanatoriumCount); 93 printSanatoriums(sanatoriums, sanatoriumCount); 94 95 string searchProfile; 96 cout << "\nВведите лечебный профиль для поиска: "; 97 cin.ignore(); 98 getline(cin, searchProfile); 99 100 searchSanatoriums(sanatoriums, sanatoriumCount, searchProfile); 101 } 102</pre>																															