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:

Студент 1 курса 10 группы

Мандрик Алексей Иванович

Преподаватель: асс. Андронова М.В.

Минск, 2024

. 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 8**



**Код программы**

#include <iostream>

#include <Windows.h>

using namespace std;

void input(int size);

void output();

void sort(double point);

typedef struct Students //Описание структуры

{

char surname[16]; //Для ввода фамилии

char name[16]; //Для ввода имени

char patronymic[16]; //Для ввода отчества

char klass[4]; //Для ввода номера класса

int subject\_1; //Для ввода отметки

int subject\_2; //Для ввода отметки

int subject\_3; //Для ввода отметки

double delta; //Для ввода среднего балла

} STUD;

int number; FILE\* f; errno\_t err;

int main()

{ // Для русского алфавита

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int choice;

double point;

// Цикл выбора нужного действия

do

{

cout << "\n1.Ввод данных с клавиатуры и запись в файл\n";

cout << "2.Вывод данных из файла\n";

cout << "3.Поиск по среднему баллу\n";

cout << "0.Выход из программы\n\n";

cout << "Введите номер операции: ";

cin >> choice;

// Вызов нужных функций пользователя

switch (choice)

{

case 1:

cout << "Введите количество учеников: ";

cin >> number;

input(number);

break;

case 2:

output();

break;

case 3: { cout << "Введите средний балл: ";

cin >> point;

sort(point);

break;

}

case 0:

exit(0);

break;

}

} while (choice != 0);

}

// Функция заполнения данными

void input(int size)

{

STUD buf = { ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '};

//Открытие файла для записи

if (!fopen\_s(&f, "base.txt", "ab"))

{

//Ввод переменных

for (int p = 0; p < size; p++)

{

cout << "Фамилия: "; cin >> buf.surname;

cout << "Имя: "; cin >> buf.name;

cout << "Отчество: "; cin >> buf.patronymic;

cout << "Класс: "; cin >> buf.klass;

cout << "Русский язык: "; cin >> buf.subject\_1;

cout << "Математика: "; cin >> buf.subject\_2;

cout << "Информатика: "; cin >> buf.subject\_3;

cout << "Средний балл: "; cin >> buf.delta;

fwrite(&buf, sizeof(buf), 1, f);

}

// Закрыть

fclose(f);

}

else {

cout << "Ошибка открытия файла";

return;

}

}

//Функция вывода данных из файла

void output()

{

STUD buf;

//Открыть для чтения

if (!fopen\_s(&f, "base.txt", "rb"))

{

cout << "\nФамилия Имя Отчество Класс Русский язык Математика Информатика Средний балл\n";

fread(&buf, sizeof(buf), 1, f);

// Вывод переменных

while (!feof(f))

{

cout << buf.surname << " " << buf.name << " " << buf.patronymic << " " << buf.klass << " " << buf.subject\_1 << " " << buf.subject\_2 << " " << buf.subject\_3 << " " << buf.delta << endl;

fread(&buf, sizeof(buf), 1, f);

}

cout << endl;

//Закрыть

fclose(f);

}

else

{

cout << "Ошибка открытия файла";

return;

}

}

// Функция поиска человека с нужным средним баллом

void sort(double point)

{

STUD buf;

// Открыть для чтения

if (!fopen\_s(&f, "base.txt", "rb"))

{

fread(&buf, sizeof(buf), 1, f);

// цикл определения нужного человека

while (!feof(f))

{

if(buf.delta == point)

cout << "Cредний балл " << buf.surname << "а равен введённому! " << endl;

fread(&buf, sizeof(buf), 1, f);

}

cout << endl;

// Закрыть

fclose(f);

}

else

{

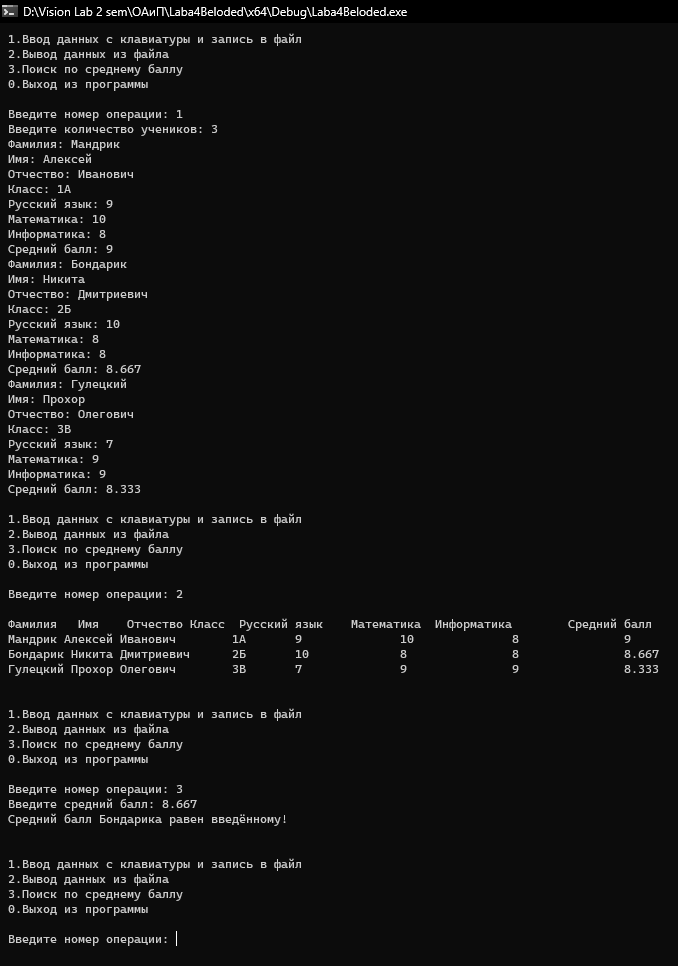
cout << "Ошибка открытия файла";

return;

}

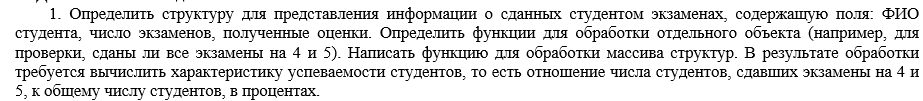
}

**Результат выполнения:**



**Дополнительные задания:**

**Задание 1**

****

**Код программы:**

#include <iostream>

#include <string>

#include <Windows.h>

using namespace std;

struct Student // СТруктура

{

string name; // Для ввода миени

int exams; // Для количества экзаменов

int\* marks; // Для отметок

};

// Функция сдан/не сдан экзамен

void check(Student\* student)

{

int count = 0;

for (int i = 0; i < student->exams; i++)

{

if (student->marks[i] >= 4 && student->marks[i] <= 10)

{

count++;

}

}

if (count == student->exams)

{

cout << "Все экзамены сданы" << endl;

}

else

{

cout << "Не все экзамены сданы" << endl;

}

}

// Функция опредления процента сдавших студентов

void checkAll(Student\* students, int size)

{

int count = 0;

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

{

int countMarks = 0;

for (int j = 0; j < students[i].exams; j++)

{

if (students[i].marks[j] >= 4 && students[i].marks[j] <= 10)

{

countMarks++;

}

}

if (countMarks == students[i].exams)

{

count++;

}

}

cout << "Процент студентов, сдавших все экзамены: " << (double)count / size \* 100 << "%" << endl;

}

int main()

{

// Для русского алфавита

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int size;

cout << "Введите количество студентов: ";

cin >> size;

Student\* students = new Student[size]; // Выделение динамической памяти для студентов

// Цикл для ввода данных

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

{

cout << "Введите имя студента: ";

cin >> students[i].name; // Ввод имени

cout << "Введите количество экзаменов: ";

cin >> students[i].exams; // Ввод количества экзаменов

students[i].marks = new int[students[i].exams]; // Выделение динамической памяти для отметок

// Цикл для ввода отметок

for (int j = 0; j < students[i].exams; j++)

{

cout << "Введите оценки: ";

cin >> students[i].marks[j];

}

}

// Цикл для вывода инфы и вызова функций пользователя

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

{

cout << "Студент: " << students[i].name << endl;

check(&students[i]); // Вызов функции check()

}

checkAll(students, size);// Вызов функции checkAll()

// Удаление динамической памяти

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

{

delete[] students[i].marks;

}

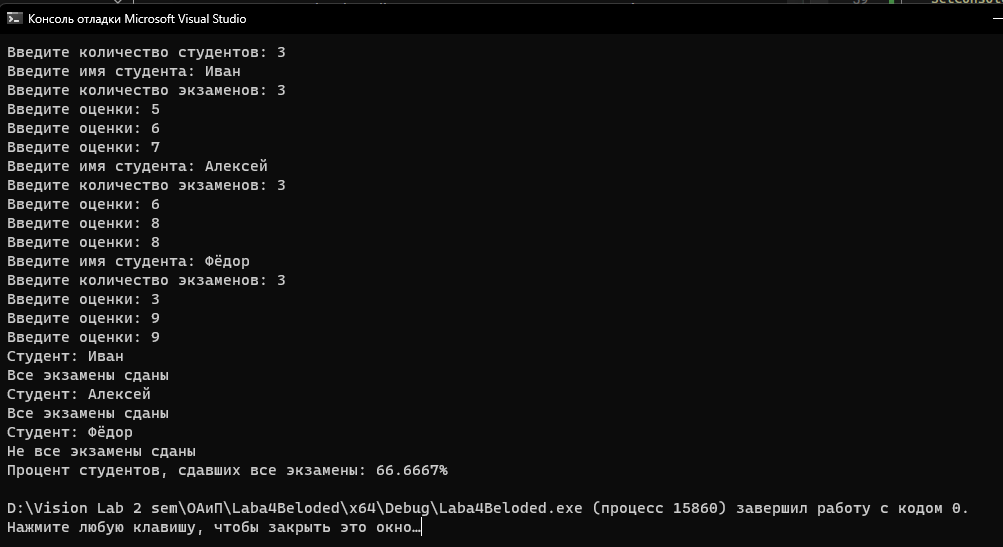
delete[] students;

return 0;

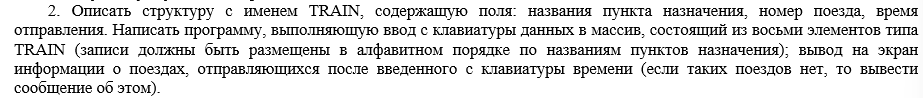
cin.get();

}

**Результат выполнения:**



**Задание 2**

****

**Код программы:**

#include <iostream>

#include <Windows.h>

# define str\_len 30

# define size 8

using namespace std;

void enter\_new(); // функция вывода

void out(); // Функция вывода

struct train // Структура

{

char Goto[str\_len];

int numb; // Для номера поезда

float time; // Для времени отправления

};

struct train train\_list[size];

int current\_size = 0;

int choice;

int main()

{

// Для русского алфавита

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

// Меню

cout << "Введите:" << endl;

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

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

cout << "3-для выхода" << endl;

cin >> choice;

do

{

switch (choice)

{

case 1:

enter\_new(); // Ввод информации

break;

case 2:

out(); // Вывод информации

break;

}

} while (choice != 3);

}

// Функция пользователя для ввода данных

void enter\_new()

{

cout << "Ввод информации" << endl;

if (current\_size < size)

{

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

cin >> train\_list[current\_size].Goto; // Ввод

cout << "Время выезда: " << endl;

cin >> train\_list[current\_size].time; // Ввод

cout << "Номер: " << endl;

cin >> train\_list[current\_size].numb; // Ввод

current\_size++;

}

else

cout << "Введено максимальное кол-во строк";

cout << "Что дальше?" << endl;

cin >> choice;

}

// Функция пользователя для вывода данных

void out()

{

float goTime;

bool atLeastOne = false;

cout << "Введите минимальное время отправления: " << endl;

cin >> goTime;

// Подбор ближайшего поезда

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

{

if (train\_list[i].time > goTime)

{

cout << endl;

cout << "Номер: " << train\_list[i].numb << endl;

cout << "Пункт назначения: " << train\_list[i].Goto << endl;

cout << "Время выезда " << train\_list[i].time << endl;

atLeastOne = true;

}

}

if (!atLeastOne) {

cout << "Таких поездов нет! " << endl;

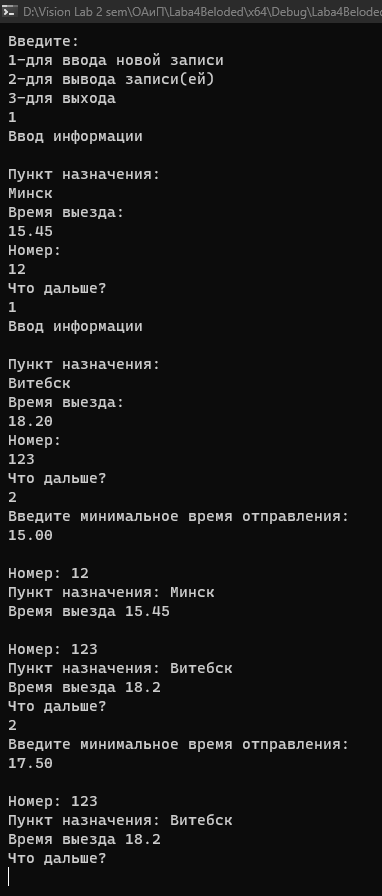
}

cout << "Что дальше?" << endl;

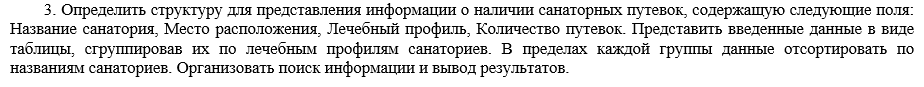
cin >> choice;

}

**Результат выполнения:**



**Задание 3**

****

**Код программы:**

#include <iostream>

#include <iomanip>

#include <string>

#include <windows.h>

using namespace std;

struct sanatorium // Структура

{

string name; // Для ввода названия

string location; // Для ввода места

string health\_profile; // Для ввода профиля

int vouchers; // Для билетов

void find\_name(string search\_name) // Функция пользователя// Функция пользователя для определённого объекта по названию

{

if (this->name == search\_name)

{

list\_output(); // Вызов функции пользователя

}

}

void find\_profile(string search\_profile) // Функция пользователя для определённого объекта по профилю

{

if (this->health\_profile == search\_profile)

{

list\_output();// Вызов функции пользователя

}

}

void list\_output() // Функция пользователя, условный шаблон вывода информации

{

cout << "Название санатория: " << name << " " << endl;

cout << "Расположение санатория: " << location << " " << endl;

cout << "Лечебный профиль: " << health\_profile << " " << endl;

cout << "Количество путёвок: " << vouchers << " " << endl;

cout << endl;

}

};

void table\_head() // Функция пользователя для информации в таблице

{

cout << left << setw(20) << "Лечебный профиль" << left << setw(20) << "Название санатория" << left

<< setw(20) << "Расположение" << left << setw(20) << "Количество путёвок" << endl << endl;

}

void output(sanatorium arr[]) // Функция пользователя для красивого вывода информации по всем санаториям

{

/\*Выравнивание с помощью выделения определённого места для каждого слова\*/

table\_head();// Функция пользователя

cout << left << setw(20) << arr[0].health\_profile

<< left << setw(20) << arr[0].name

<< left << setw(20) << arr[0].location

<< left << setw(20) << arr[0].vouchers

<< "\n\n\n\n";

cout << left << setw(20) << arr[2].health\_profile

<< left << setw(20) << arr[2].name

<< left << setw(20) << arr[2].location

<< left << setw(20) << arr[2].vouchers

<< "\n\n\n\n";

table\_head();// Функция пользователя

cout << left << setw(20) << arr[1].health\_profile

<< left << setw(20) << arr[1].name

<< left << setw(20) << arr[1].location

<< left << setw(20) << arr[1].vouchers

<< "\n\n\n\n";

table\_head();// Функция пользователя

cout << left << setw(20) << arr[4].health\_profile

<< left << setw(20) << arr[4].name

<< left << setw(20) << arr[4].location

<< left << setw(20) << arr[4].vouchers

<< "\n\n\n\n";

table\_head();// Функция пользователя

cout << left << setw(20) << arr[3].health\_profile

<< left << setw(20) << arr[3].name

<< left << setw(20) << arr[3].location

<< left << setw(20) << arr[3].vouchers

<< "\n\n\n\n";

}

int main() {

// Для русского алфавита

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

sanatorium objects[5]; // 5 примеорв

objects[0].name = "Березка"; objects[0].location = "Солигроск"; objects[0].health\_profile = "Осанка"; objects[0].vouchers = 155;

objects[1].name = "Росинка"; objects[1].location = "д.Мурашки"; objects[1].health\_profile = "Осанка"; objects[1].vouchers = 247;

objects[2].name = "Паласа"; objects[2].location = "д.Париж"; objects[2].health\_profile = "Зависимости"; objects[2].vouchers = 102;

objects[3].name = "Свитязь"; objects[3].location = "Минск"; objects[3].health\_profile = "Опорный аппарат"; objects[3].vouchers = 20;

objects[4].name = "Ивязь"; objects[4].location = "Слуцк"; objects[4].health\_profile = "Зрение"; objects[4].vouchers = 6;

int choice;

while (true)

{

// Меню

cout << "\t" << "Введите нужный вариант" << endl;

cout << "\n1 - Поиск информации по названию";

cout << "\n2 - Поиск информации по лечебному профилю";

cout << "\n3 - Вывод всех существующих санаториев";

cout << "\n4 - Выход из программы" << endl;

cin >> choice;

//Выбор нужного варианта

switch (choice)

{

case 1: {

string name;

cout << "Введите название санатория...";

cin.ignore();

getline(cin, name);

// Вывод информации по конкретному санаторию

for (int i = 0; i < 5; i++)

{

objects[i].find\_name(name); // Функция пользователя для определённого объекта

}

break;

}

case 2: {

string name;

cout << "Введите лечебный профиль санатория...";

cin.ignore();

getline(cin, name);

// Вывод информации по конкретным санаториям с определённым лечебным профилем

for (int i = 0; i < 5; i++)

{

objects[i].find\_profile(name); // Функция пользователя для определённого объекта

}

break;

}

case 3: {

output(objects); // Вызов функции пользователя

cout << endl << endl;

break;

}

default:

break;

}

}

}

**Результат выполнения:**

