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»

 Выполнила:

Студентка1 курса 6 группы

Альшевская Алина Михайловна

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 1

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 <fstream>

#include <string>

using namespace std;

void input(int size);

void output();

void find(char date[11]);

void del();

void filewriting();

void filereading();

struct Cityzens

{

char fio[50];

char date[11];

char adres[50];

char sex;

};

struct Cityzens list\_of\_people[30];

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

int main()

{

setlocale(LC\_ALL, "Russian");

int choice; char date[11];

do

{

cout << "\n1.Ввод данных с клавиатуры \n";

cout << "2.Вывод данных \n";

cout << "3.Поиск по году рождения\n";

cout << "4.Удаление строки\n";

cout << "5. Запись информации в файл\n";

cout << "6. чтение данных из файла\n";

cout << "0.Выход из программы\n\n";

cout << "Введите номер операции: ";

cin >> choice;

switch (choice) //для выбора действия

{

case 1: {cout << "Введите количество жителей: ";

cin >> number;

input(number); break;

}

case 2: {output(); break; }

case 3: { cout << "Введите дату рождения: ";

cin >> date;

find(date); break;

}

case 4: {

del(); break;

};

case 5: {filewriting(); break; };

case 6: {filereading(); break; };

case 0: exit(0); break;

}

} while (choice != 0);

}

void input(int size) //функция ввода данных

{

for (int p = 0; p < size; p++)//используем цикл для заполнения

{

cout << "\nФИО "; cin >> list\_of\_people[p].fio;

cout << "дата рождения ( в формате дд.мм.гггг) "; cin >> list\_of\_people[p].date;

cout << "Адрес(Город, улица, дом, квартира) "; cin >> list\_of\_people[p].adres;

cout << "Пол (м/ж) "; cin >> list\_of\_people[p].sex;

}

}

void output( ) //функция вывода данных

{

for(int i=0;i<number;i++)//используем цикл для вывода

{

cout << "\nФИО " << list\_of\_people[i].fio << endl;

cout << "дата рождения ( в формате дд.мм.гггг) "<< list\_of\_people[i].date << endl;

cout << "Адрес(Город, улица, дом, квартира)"<< list\_of\_people[i].adres <<endl;

cout << "Пол (м/ж)" << list\_of\_people[i].sex << endl;

}

}

void del()//функция удаления строк

{

int d;

cout << "\nНомер строки, которую надо удалить " << endl; cin >> d;

for (int de1 = (d - 1); de1 < number-1; de1++) {

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

}

number = number - 1;

}

void find(char date[10]) //функция поиска по дате

{

bool flag = false;

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

if (strcmp(date, list\_of\_people[i].date) == 0) //сравнение строк

{

cout << "ФИО " << list\_of\_people[i].fio << endl;

cout << "дата рождения ( в формате дд.мм.гггг) " << list\_of\_people[i].date << endl;

cout << "Адрес(Город, улица, дом, квартира)" << list\_of\_people[i].adres << endl;

cout << "Пол (м/ж)" << list\_of\_people[i].sex << endl;

flag = true; break;

}

if (!flag) cout << "Ничего не найдено\n";

}

void filewriting() {// функция записи в файл

ofstream file("data.txt");

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

{

file<< "\nФИО " << list\_of\_people[i].fio << endl;

file << "дата рождения ( в формате дд.мм.гггг) " << list\_of\_people[i].date << endl;

file << "Адрес(Город, улица, дом, квартира)" << list\_of\_people[i].adres << endl;

file << "Пол (м/ж)" << list\_of\_people[i].sex << endl;

}

file.close();

}

void filereading() {// функция чтения файла

ifstream file("data.txt");

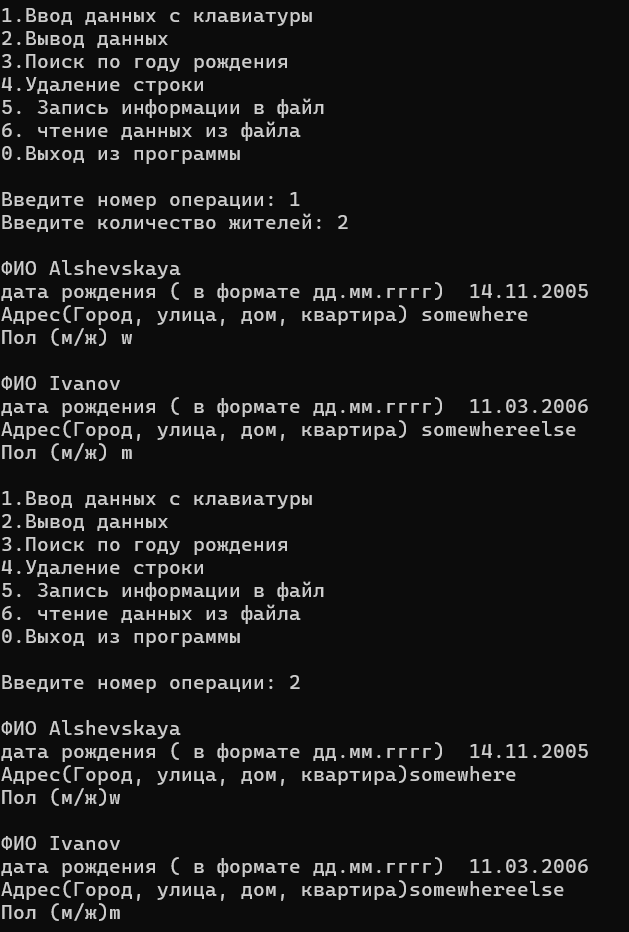
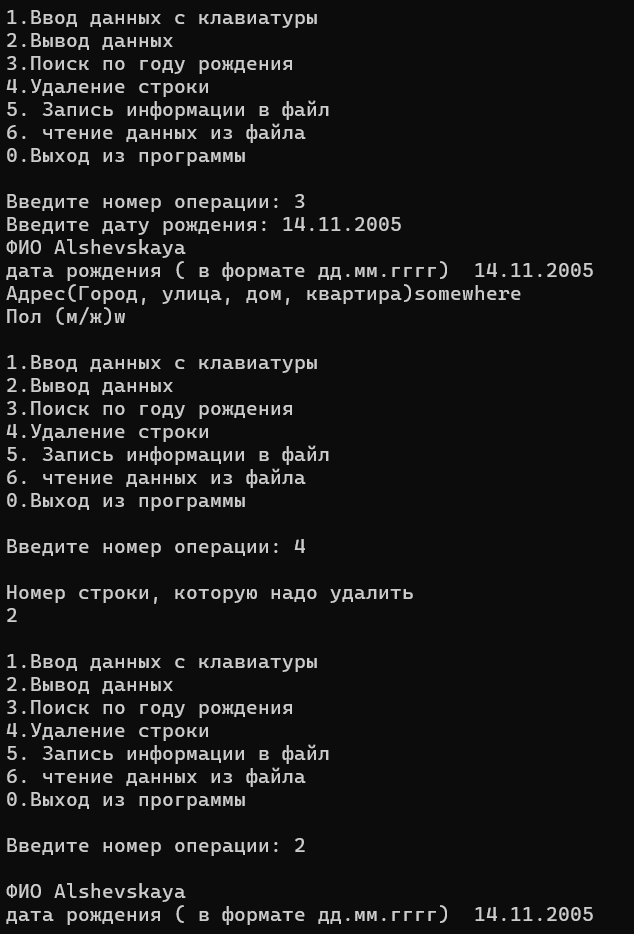
string line;

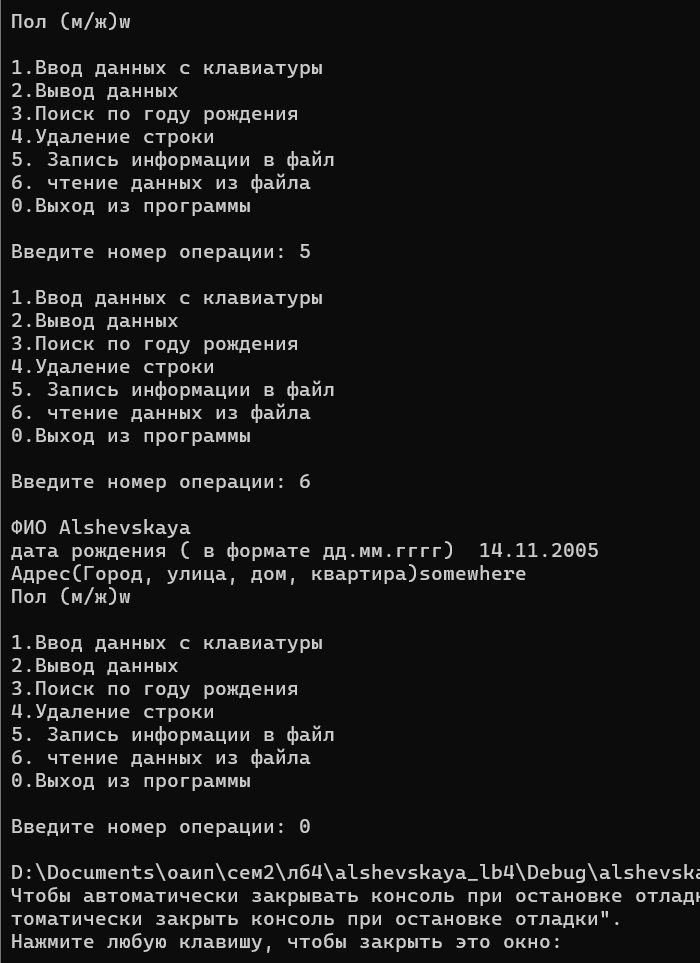
while (getline(file, line)) {

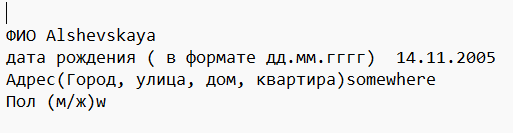
cout << line << endl;

}

file.close();

}





Дополнительные задания

Задание 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.

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct students

{

char fio[50];

int amm\_of\_exems;

int masofg[30];

};

struct students mas[30];

int number;

int main()

{

setlocale(LC\_ALL, "Russian");

cout << "Input number of students ";

cin >> number;

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

{

cout << "\ninput fio ";//вводим фио студента

cin >> mas[i].fio;

cout << "input ammount of exems ";//ввод количества экзаменов

cin >> mas[i].amm\_of\_exems;

for (int j = 0; j < mas[i].amm\_of\_exems; j++) {//ввод оценок за экзамены

cout << "input grade ";

cin >> mas[i].masofg[j];

}

}

int counter\_of\_stud = 0;

for (int i = 0; i < number; i++)//используя циул считает оуенки каждого ученика

{

int counter = 0;

for (int j = 0; j < mas[i].amm\_of\_exems; j++) {

if (mas[i].masofg[j] == 4 || mas[i].masofg[j] == 5)// если оценка 4 или 5 то учеличиваем счётчик

counter++;

}

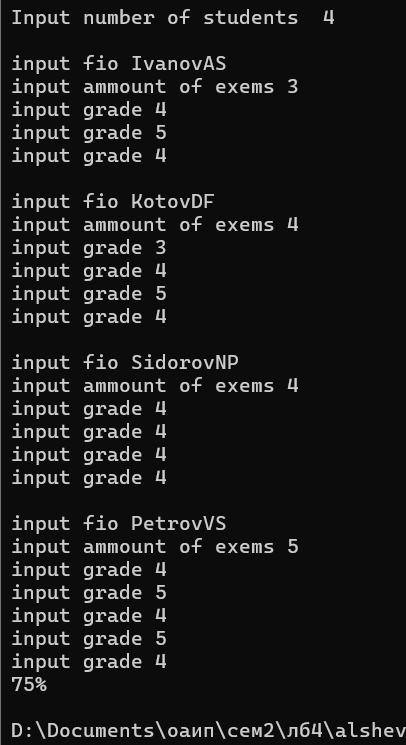
if (counter == mas[i].amm\_of\_exems) counter\_of\_stud++;// если счётчик = кол-ву экзаменов, то увеличиваеи счётчик судентов

}

float proc = counter\_of\_stud\*100 / number;//студенты с 4 и 5/ кол-во всех студ и на 100 умножить

cout <<proc << "%" << endl;

}



Задание 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).

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct TRAIN

{

string mesto;;

int numberoftrain;

string time;

};

struct TRAIN mas[8];

int main()

{

setlocale(LC\_ALL, "Russian");

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

{

cout << "\ninput place you arrive to ";//вводим место приытия

cin >> mas[i].mesto;

cout << "input number of train ";//ввод номер поещда

cin >> mas[i].numberoftrain;

cout<< "Input time like 00:00 ";// Вводим время приытия

cin >> mas[i].time;

}

TRAIN temp;//создаем временную структуру

for (int i = 0; i < 8; i++) {//цикл для двойной сортировки

for (int j = 0; j < 8; j++) {

if (mas[j].mesto > mas[j + 1].mesto) {

temp = mas[j];

mas[j] = mas[j + 1];

mas[j + 1] = temp;

}

}

}

for (int i = 1; i < 10; i++)//вывод отсортированного массива

{

cout << "\nPLace ";

cout << mas[i].mesto << endl;

cout << "Number of train ";

cout << mas[i].numberoftrain << endl;

cout << "time ";

cout << mas[i].time << endl;

}

string departure\_time;//ввод времени для поиска

cout << "Enter the departure time to see nearest trains: ";

cin>> departure\_time;

string hours;// переводим часы в int

for (int i = 0; i < 2; i++) {

hours += departure\_time[i];

}

int hours\_num = stoi(hours);

string minutes;//переводим минуты в int

for (int i = 3; i < 5; i++) {

minutes += departure\_time[i];

}

int minutes\_num = stoi(minutes);

bool check\_near = false;

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

//переводим часы прибытия в int

string check\_hours;

for (int j = 0; j < 2; j++) {

check\_hours += mas[i].time[j];

}

int check\_hours\_num = stoi(check\_hours);

if (check\_hours\_num >= hours\_num && check\_hours\_num <= hours\_num + 1) {// проверяем есть ли поезд в теч часа

check\_near = true;

//если есть то выдодим информацию про него

cout << "\n time " << mas[i].time << endl;

cout << "place " << mas[i].mesto << endl;

cout << "number of train " << mas[i].numberoftrain << endl;

cout << endl;

}

}

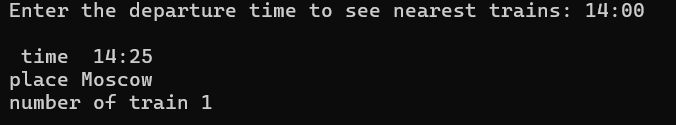
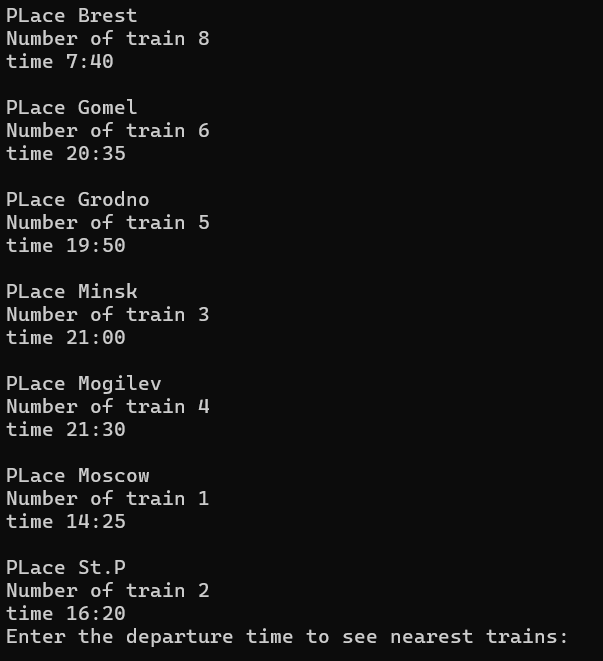
//если нету

if (check\_near == false) {

cout << "Unfortunately, there are no trains departing within at least 1 hour." << endl;

}

}



Задание 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.

#include <iostream>

using namespace std;

struct Sanatorium {

string name;

string location;

string profile;

int quantity;

};

void printSanatorium(const Sanatorium& sanatorium) {

std::cout << sanatorium.name << "\t" << sanatorium.location << " " << sanatorium.profile << " " << sanatorium.quantity << std::endl;

}

int main() {

const int MAX\_SANATORIUMS = 5;

Sanatorium sanatoriums[MAX\_SANATORIUMS] = {

{"Relax ", "Belarus", "joints ", 1},

{"Holidays", "Russia ", "lungs ", 3},

{"Family ", "Italy ", "legs ", 15},

{"InForest", "Filand ", "nevrology", 11},

{"Green ", "Filand ", "heart ", 18}

};

// сортируем сонатории по лечебным профилям

for (int i = 0; i < MAX\_SANATORIUMS - 1; i++) {

for (int j = 0; j < MAX\_SANATORIUMS - i - 1; j++) {

if (sanatoriums[j].profile > sanatoriums[j + 1].profile) {

Sanatorium temp = sanatoriums[j];

sanatoriums[j] = sanatoriums[j + 1];

sanatoriums[j + 1] = temp;

}

}

}

// Pвывод отсортированой структуры

std::cout << "Name Location Profile Quantity" << std::endl;

for (int i = 0; i < MAX\_SANATORIUMS; i++) {

printSanatorium(sanatoriums[i]);

}

// посик санотория по имени

std::string searchName;

std::cout << "\nEnter the name of the sanatorium you want to search for: ";

std::cin >> searchName;

bool found = false;

for (int i = 0; i < MAX\_SANATORIUMS; i++) {

if (sanatoriums[i].name == searchName) {

std::cout << "Sanatorium found:" << std::endl;

printSanatorium(sanatoriums[i]);

found = true;

break;

}

}

if (!found) {

std::cout << "Sanatorium not found." << std::endl;

}

return 0;

}