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»

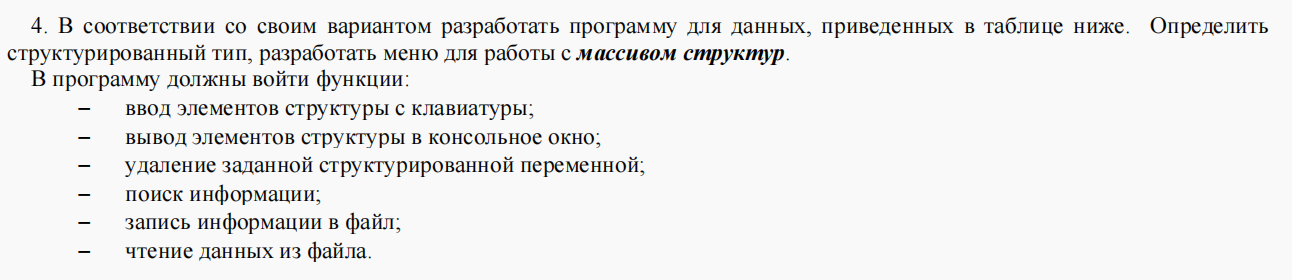
Выполнил:

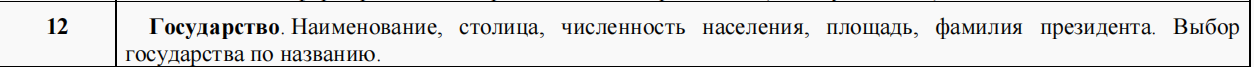
Студент 1 курса 6 группы

Романов Игорь Вячеславович

Преподаватель: асс. Андронова М.В.

2024, Минск





#include <iostream>

#include <Windows.h>

#include <iostream>

#include <fstream>

#define LIMIT 100 // максимальное количество стран для массива структур

//прототипы функций

using namespace std;

void inputInfo();

void outputInfo();

void searchInfo();

void deleteCountry();

void writeInfo();

void readInfo();

// структура

struct Goverment

{

char name[16];

char capital[16];

unsigned int population;

unsigned int area;

char president[16];

};

Goverment country[LIMIT]; //массив country из объектов country

int current\_size = 0; // текущий размер массива

int main()

{

SetConsoleOutputCP(1251); // для распознавания кириллицы при записи в массив

SetConsoleCP(1251);

int choice;

do {

cout << "\nВведите вариант работы приложения: \n";

cout << "0. Выход из приложения\n";

cout << "1. Ввод элементов структуры с клавиатуры\n";

cout << "2. Вывод элементов структуры в консольное окно\n";

cout << "3. Удаление заданной структурированной переменной\n";

cout << "4. Поиск информации\n";

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

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

cin >> choice;

switch (choice)

{

case 1: { //Ввод элементов структуры с клавиатуры

inputInfo();

break;

}

case 2: { //Вывод элементов структуры в консольное окно

outputInfo();

break;

}

case 3: { //Удаление заданной структурированной переменной

deleteCountry();

break;

}

case 4: { //Поиск информации

searchInfo();

break;

}

case 5: { //Запись информации в файл

writeInfo();

break;

}

case 6: { //Чтение данных из файла

readInfo();

break;

}

case 0: { //Выход из программы

exit(0);

break;

}

default: {

cout << "Введено некорректное значение. " << endl;

break;

}

}

} while (choice != 0);

}

void inputInfo()

{

if (current\_size < LIMIT) { //если массив не выходит за пределы заданного, то вводим соотв. информацию

cout << "Введите название: " << endl;

cin >> country[current\_size].name;

cout << "Введите столицу: " << endl;

cin >> country[current\_size].capital;

cout << "Введите численность населения: " << endl;

cin >> country[current\_size].population;

cout << "Введите площадь территории (в км^2): " << endl;

cin >> country[current\_size].area;

cout << "Введите фамилию президента страны: " << endl;

cin >> country[current\_size].president;

current\_size++;

cout << "Элемент структуры создан. " << endl;

}

else { //иначе выводим сообщение о переполнении

cout << "Структура переполнена!" << endl;

}

}

void outputInfo()

{

for (int i = 0; i < current\_size; i++) { //поочередно выводим информацию о странах

cout << "\nСтрана №" << i + 1 << endl;

cout << "Название: " << country[i].name << endl;

cout << "Столица: " << country[i].capital << endl;

cout << "Численность населения: " << country[i].population << endl;

cout << "Площадь страны: " << country[i].area << endl;

cout << "Президент страны: " << country[i].president << endl;

}

}

void searchInfo() {

char country\_name[20]; //переменная для поиска страны

cout << "По какой стране вы хотите найти информацию? " << endl;

cin >> country\_name;

for (int i = 0; i < 1; i++) {

if (!strcmp(country\_name, country[i].name)) { //если введенная страна найдена, то выводим информацию о ней

cout << "Информация о выбранной стране: " << country\_name << endl;

cout << "Столица: " << country[i].capital << endl;

cout << "Численность населения: " << country[i].population << endl;

cout << "Площадь страны: " << country[i].area << endl;

cout << "Президент страны: " << country[i].president << endl;

}

else { //иначе выводим сообщение об ошибке

cout << "Страна была введена неверно" << endl;

}

}

}

void deleteCountry() {

char country\_name[20]; //переменная для поиска страны

cout << "Введите страну, которую хотите удалить из структуры: " << endl;

cin >> country\_name;

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

if (!strcmp(country\_name, country[i].name)) { //если введенная страна найдена, то удаляем ее из структуры

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

country[j] = country[j + 1];

}

current\_size--;

cout << "Структурная переменная очищена. " << endl;

}

else { //иначе выводим сообщение об ошибке

cout << "Страна не была найдена" << endl;

}

}

}

void writeInfo() {

ofstream ofile("countries.txt"); // создание объекта класса ofstream для записи

if (ofile.fail()) { // в случае ошибки открытия файла

cout << "Файл не удалось открыть!" << endl;

exit(1);

}

for (int i = 0; i < current\_size; i++) { //записываем информацию в файл

ofile << country[i].name << " " << country[i].capital << " " << country[i].population << " " << country[i].area << " " << country[i].president << " " << endl;

}

ofile.close(); //закрытие файла

cout << "Информация о странах успешна записана в файл";

}

void readInfo() {

ifstream ifile("countries.txt"); // создание объекста класса ifstream для чтения

if (ifile.fail()) { // в случае ошибки открытия файла

cout << "Файл не удалось открыть!" << endl;

exit(1);

}

current\_size++; //увеличиваем размера массива для будущей записи страны

for (int i = 0; i < current\_size; i++) { //читаем информацию о тсране из файла

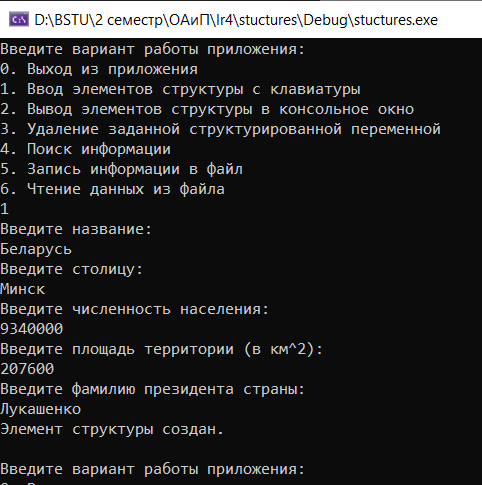
ifile >> country[i].name >> country[i].capital >> country[i].population >> country[i].area >> country[i].president;

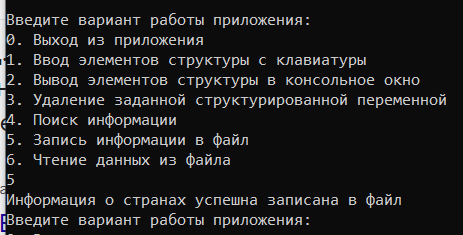
}

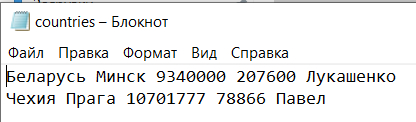
ifile.close(); //закрытие файла

cout << "Информация из файла успешно прочитана и записана в структуру";

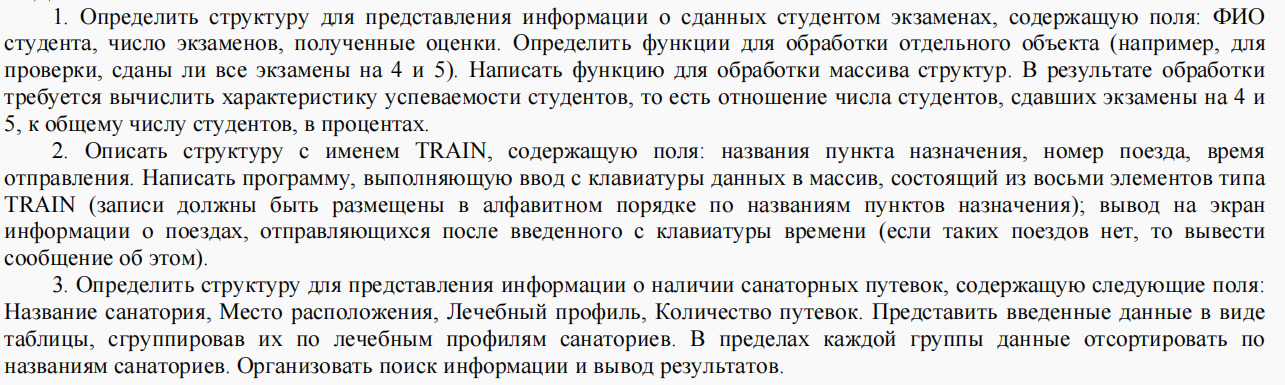
}







Доп. задания:

1.  


#include <iostream>

#include <Windows.h>

//прототипы функций

using namespace std;

bool isExamPassed(int);

void getStatistics();

// структура

struct Marks

{

char FIO[80]; //фио ученика

const int amount = 3; //количество эзкаменов

int marks[3]; //массив с оценкам

};

//определяем структуру для студентов и их оценками по экзамену

const int students = 3;

Marks exam\_list[students] = {

{"Student1", 3, {4, 5, 4}},

{"Student2", 3, {5, 5, 3}},

{"Student3", 3, {4, 4, 4}}

};

// главная функция

int main()

{

SetConsoleOutputCP(1251); // для распознавания кириллицы при записи в массив

SetConsoleCP(1251);

//определяем, сдали ли студенты все экзамены

for (int i = 0; i < exam\_list[i].amount; i++) {

if (isExamPassed(i)) {

cout << "Студент №" << i + 1 << " сдал все экзамены" << endl;

}

else {

cout << "Студент №" << i + 1 << " сдал НЕ все экзамены" << endl;

}

}

// получаем статистику по предмету

getStatistics();

}

// фукнция для проверки сдачи всех экзаменов студентом

bool isExamPassed(int num){

for (int j = 0; j < exam\_list[num].amount; j++) { //цикл для обработки массива в массиве структур

if (exam\_list[num].marks[j] < 4) { //если найдена оценка меньше 4

return false;

}

}

return true;

}

// функция для сбора статистику по предмету

void getStatistics() {

for (int i = 0; i < exam\_list[i].amount; i++) {

double result\_4 = 0, result\_5 = 0; // создаем счетчики для подсчета отношений

for (int j = 0; j < exam\_list[i].amount; j++) {

if (exam\_list[j].marks[i] == 4) {

result\_4++;

}

if (exam\_list[j].marks[i] == 5) {

result\_5++;

}

}

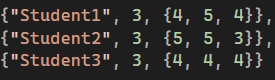
//выводим соотв. сообщения

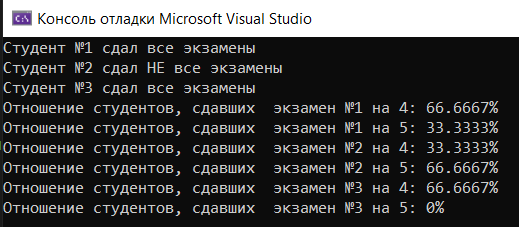
cout << "Отношение студентов, сдавших экзамен №" << i + 1 << " на 4: " << (result\_4 / students) \* 100 << "%" << endl;

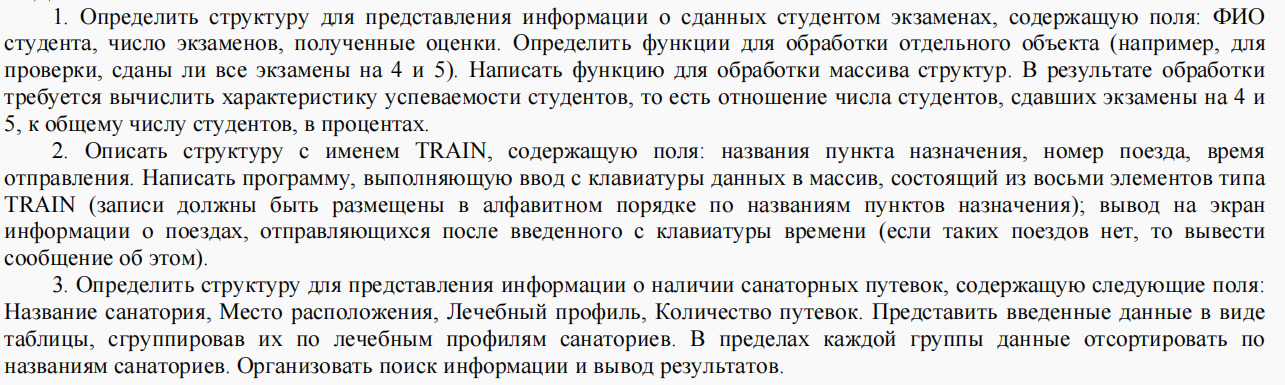
cout << "Отношение студентов, сдавших экзамен №" << i + 1 << " на 5: " << (result\_5 / students) \* 100 << "%" << endl;

}

}





2. 

#include <iostream>

#include <Windows.h>

using namespace std;

// структура

struct TRAIN

{

char destination[80]; // пункт назначения

unsigned int train\_number; //номер поезда

char arrival\_time[80]; //время прибытия

};

// прототипы функций

void sortStructures(TRAIN train[], int num);

void compareTime(TRAIN train[], char ask\_time[]);

//определяем структуру для поездов и информации и о них

TRAIN trains[8];

// главная функция

int main()

{

SetConsoleOutputCP(1251); // для распознавания кириллицы при записи в массив

SetConsoleCP(1251);

char ask\_time[8]; //время, которое запрашивает пользователь

//ввод данных в массив структуры

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

cout << "Поезд №" << i + 1 << ": " << endl;

cout << "Введите пункт назначения: ";

cin >> trains[i].destination;

cout << "Введите номер поезда: ";

cin >> trains[i].train\_number;

cout << "Введите время прибытия в формате ЧЧ:ММ: ";

cin >> trains[i].arrival\_time;

}

sortStructures(trains, 8); //сортируем элементы

cout << "Введите интересующее вас время в формате ЧЧ:ММ " << endl;

cin >> ask\_time;

compareTime(trains, ask\_time); //сравниваем время прибытия и введенное значение

return 0;

}

// функция для сортировку поездов по алфавиту

void sortStructures(TRAIN train[], int num) {

for (int i = 0; i < num - 1; i++) {

for (int j = 0; j < num - i - 1; j++) {

if (strcmp(train[j].destination, train[j + 1].destination) > 0) { // сравниваем строки пунктов назначения

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

}

}

}

}

// фукнция для сравнения времен

void compareTime(TRAIN train[], char ask\_time[]) {

bool flag = false; // по умолчанию считается, что поездов с подходящим временем не нашлось

for (int i = 0; i < 8; i++) { //сравнвиваем посимвольно время

if (strcmp(train[i].arrival\_time, ask\_time) > 0) { //если время поездов больше введенного времени

cout << "Вам подходит: " << endl; // выводим сообщение

cout << "Поезд №" << trains[i].train\_number << ". Пункт назначения: " << trains[i].destination << " . Время прибытия: " << trains[i].arrival\_time << endl;

flag = true; //устанавливаем флажок, что нашли

}

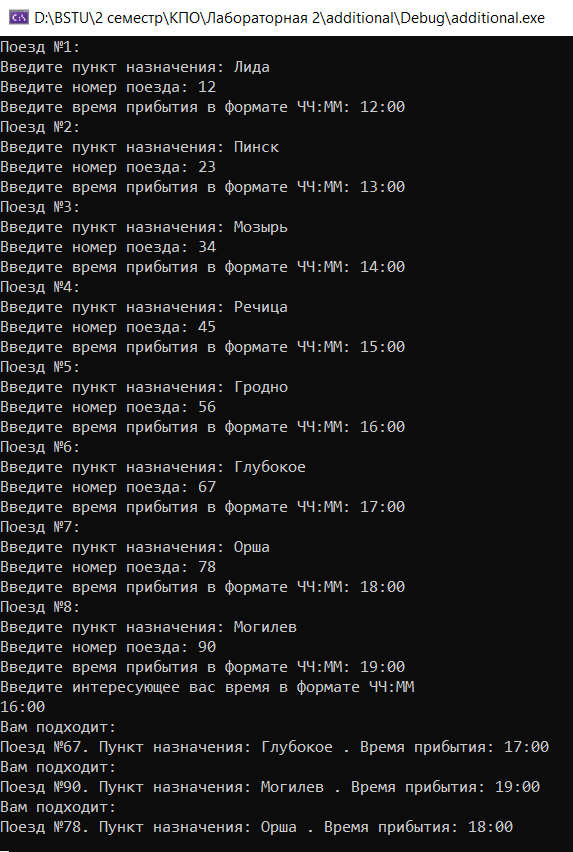
}

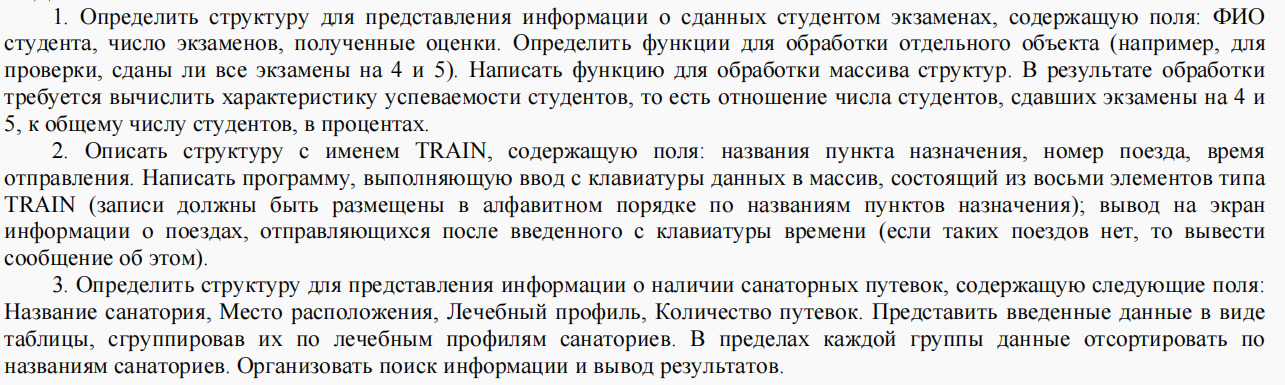
if (!flag) { //если время поездов меньше введенного времени

cout << "Поездов с подходящим временем не найдено" << endl;

}

}



3. 

#include <iostream>

#include <Windows.h>

using namespace std;

// структура

struct SANATORIUM

{

char name[40]; // название санатория

char location[40]; // местоположение

char profile[40]; //профиль

unsigned int amount\_package; //количество путевок

};

// прототипы функций

void sortStructures(SANATORIUM sanat[], int num);

void findSanat(SANATORIUM sanat[], char ask\_sanat[]);

void outputSanatoriums(SANATORIUM sanat[], int num);

//определяем структуру для санаториев и информации и о них

SANATORIUM sanat[5] = {

{"Пралеска", "Беларусь", "Профиль A", 13},

{"Зеленый бор", "Беларусь", "Профиль Б", 7},

{"Нарочь", "Беларусь", "Профиль A", 2},

{"Криница", "Беларусь", "Профиль A", 9},

{"Буг", "Беларусь", "Профиль Б", 6}

};

// главная функция

int main()

{

SetConsoleOutputCP(1251); // для распознавания кириллицы при записи в массив

SetConsoleCP(1251);

char ask\_sanat[40]; // переменная для запроса профиля

//ввод данных в массив структуры

sortStructures(sanat, 5); //сортируем элементы

outputSanatoriums(sanat, 5);

cout << "Какой санаторий вас интересует? " << endl;

cin >> ask\_sanat;

findSanat(sanat, ask\_sanat); //ищем нужный санаторий

return 0;

}

// функция для сортировки санаториев по алфавиту

void sortStructures(SANATORIUM sanat[], int num) {

for (int i = 0; i < num - 1; i++) {

for (int j = 0; j < num - i - 1; j++) {

if (strcmp(sanat[j].name, sanat[j + 1].name) > 0) { // сравниваем строки названий санаториев

swap(sanat[j].name, sanat[j + 1].name);

}

}

}

}

//вывод санаториев в виде таблицы

void outputSanatoriums(SANATORIUM sanat[], int num) {

// первая строка таблицы

cout << "Лечебный профиль" << "\t \t";

cout << "Название" << "\t \t";

cout << "Местоположение" << endl;

// вторая строка таблицы

cout << "Санатории по профилю А" << "\t \t";

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

if (strcmp(sanat[i].profile, "Профиль A") == 0) {

cout << sanat[i].name << "\t \t \t";

cout << sanat[i].location << endl << "\t \t \t \t";

}

}

// третья строка таблицы

cout << "\rСанатории по профилю Б" << "\t \t";

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

if (strcmp(sanat[i].profile, "Профиль Б") == 0) {

cout << sanat[i].name << "\t \t";

cout << sanat[i].location << endl << "\t \t \t \t";

}

}

cout << "\r"; //возвращение каретки в начало строки

}

// поиск путевок по названию санатория

void findSanat(SANATORIUM sanat[], char ask\_sanat[]) {

bool found = false; // флаг для отслеживания, был ли найден санаторий

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

if (strcmp(sanat[i].name, ask\_sanat) == 0) {

cout << "Количество доступных путевок: " << sanat[i].amount\_package << endl;

found = true;

break; // если нашли санаторий, то выходим из цикла

}

}

if (!found) { // если санаторий не был найден

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

}

}

