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 2 группы  
Махаров Эшгин Илхамович

Преподаватель: асс. Андронова М.В.

Минск, 2024

ВАРИАНТ 5

В соответствии со своим вариантом разработать программу для данных, приведенных в таблице ниже. Определить структурированный тип, разработать меню для работы с ***массивом структур***.

**Ломбард.** База хранимых товаров и недвижимости: анкетные данные клиента, наименование товара, оценочная стоимость; сумма, выданная под залог, дата сдачи, срок хранения. Выбор по наименованию товара.

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

// Определение структурированного типа "Товар"

struct Item {

int clientId;

string itemName;

double estimatedValue;

double pledgedAmount;

string deliveryDate;

string storagePeriod;

};

const int MAX\_ITEMS = 100; // Максимальное колво товаров

// Ввод элементов структуры с клавиатуры

void inputItem(Item& item) {

cout << "Введите ID клиента: ";

cin >> item.clientId;

cout << "Введите название товара: ";

cin >> item.itemName;

cout << "Введите оценочную стоимость: ";

cin >> item.estimatedValue;

cout << "Введите сумму, выданную под залог: ";

cin >> item.pledgedAmount;

cout << "Введите дату сдачи: ";

cin >> item.deliveryDate;

cout << "Введите срок хранения: ";

cin >> item.storagePeriod;

}

// Вывод элементов структуры в консольное окно

void outputItem(const Item& item) {

cout << "ID клиента: " << item.clientId << ", Название товара: " << item.itemName << ", Оценочная стоимость: " << item.estimatedValue << ", Сумма, выданная под залог: " << item.pledgedAmount << ", Дата сдачи: " << item.deliveryDate << ", Срок хранения: " << item.storagePeriod << endl;

}

// Удаление заданной переменной

void deleteItem(Item\* items, int& itemCount, int index) {

if (index < itemCount && index >= 0) {

for (int i = index; i < itemCount - 1; i++) {

items[i] = items[i + 1];

}

itemCount--;

cout << "Элемент удален" << endl;

}

else {

cout << "Некорректный индекс" << endl;

}

}

// Поиск информации

void searchItem(const Item\* items, int itemCount, const string& itemName) {

bool found = false;

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

if (items[i].itemName == itemName) {

outputItem(items[i]);

found = true;

}

}

if (!found) {

cout << "Товар с таким наименованием не найден" << endl;

}

}

// Запись информации в файл

void writeToFile(const Item\* items, int itemCount, const string& filename) {

ofstream file(filename);

if (file.is\_open()) {

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

file << items[i].clientId << " " << items[i].itemName << " " << items[i].estimatedValue << " " << items[i].pledgedAmount << " " << items[i].deliveryDate << " " << items[i].storagePeriod << endl;

}

file.close();

cout << "Данные успешно записаны в файл" << endl;

}

else {

cout << "Не удалось открыть файл для записи" << endl;

}

}

// Чтение данных из файла

void readFromFile(Item\* items, int& itemCount, const string& filename) {

ifstream file(filename);

if (file.is\_open()) {

while (itemCount < MAX\_ITEMS && file >> items[itemCount].clientId >> items[itemCount].itemName >> items[itemCount].estimatedValue >> items[itemCount].pledgedAmount >> items[itemCount].deliveryDate >> items[itemCount].storagePeriod) {

itemCount++;

}

file.close();

cout << "Данные успешно прочитаны из файла" << endl;

}

else {

cout << "Не удалось открыть файл для чтения" << endl;

}

}

int main() {

setlocale(0, "");

string searchName;

setlocale(0, "");

Item items[MAX\_ITEMS]; // Массив структурный

int itemCount = 0; // Счетчик коливо элементов

int choice;

do {

cout << "1. Ввести новый товар" << endl;

cout << "2. Вывести список товаров" << endl;

cout << "3. Удалить товар" << endl;

cout << "4. Поиск по наименованию товара" << endl;

cout << "5. Записать данные в файл" << endl;

cout << "6. Прочитать данные из файла" << endl;

cout << "7. Выйти" << endl;

cout << "Выберите действие: ";

cin >> choice;

switch (choice) {

case 1:

if (itemCount < MAX\_ITEMS) {

inputItem(items[itemCount]);

itemCount++;

}

else {

cout << "Массив полон, новые товары не могут быть добавлены" << endl;

}

break;

case 2:

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

outputItem(items[i]);

}

break;

case 3:

int index;

cout << "Введите индекс товара для удаления: ";

cin >> index;

deleteItem(items, itemCount, index);

break;

case 4:

cout << "Введите наименование товара для поиска: ";

cin >> searchName;

searchItem(items, itemCount, searchName);

break;

case 5:

writeToFile(items, itemCount, "cheto.txt");

break;

case 6:

readFromFile(items, itemCount, "cheto.txt");

break;

case 7:

cout << "Выход из программы" << endl;

break;

default:

cout << "Некорректный выбор" << endl;

break;

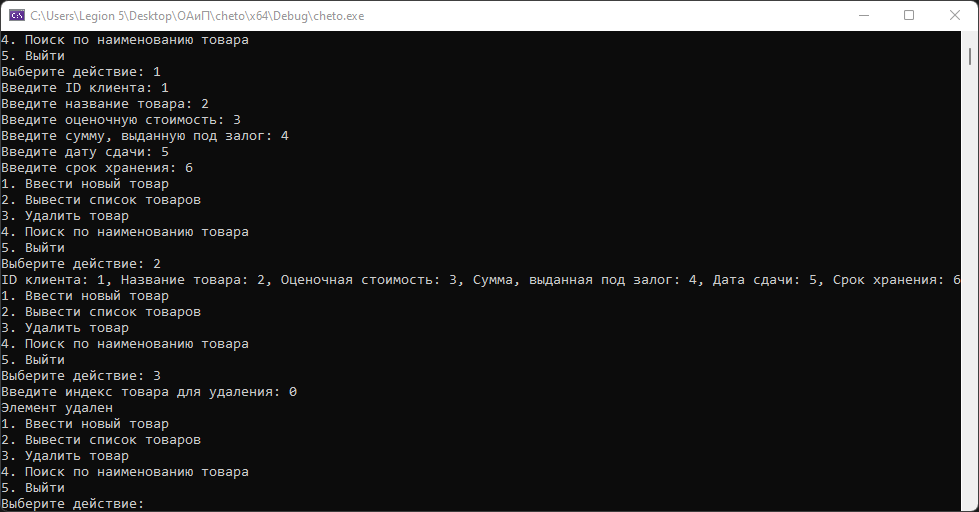
}

} while (choice != 7);

return 0;

}

Единственное недочет: для удаления товара требуется помнить его индекс. Подсчет начинать с 0.



Доп. Задачи

1. 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 <string>

#include <algorithm>

using namespace std;

// Описание структуры TRAIN

struct TRAIN {

string destination; // Пункт остановки

int trainNumber;

string departureTime; // Время отправки

};

int main() {

TRAIN trains[8];

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

cout << "Введите название пункта назначения: ";

cin >> trains[i].destination;

cout << "Введите номер поезда: ";

cin >> trains[i].trainNumber;

cout << "Введите время отправления: ";

cin >> trains[i].departureTime;

// Сортировка по названию пункта назначения

sort(trains, trains + 8, [](const TRAIN &a, const TRAIN &b) {

return a.destination < b.destination;

});

}

string inputTime;

cout << "Введите время для поиска: ";

cin >> inputTime;

bool found = false; // Флаг наличия поездов с заданным временем

// Вывод информации о поездах после введенного времени

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

if (trains[i].departureTime > inputTime) {

cout << "Пункт назначения: " << trains[i].destination << ", Номер поезда: " << trains[i].trainNumber << ", Время отправления: " << trains[i].departureTime << std::endl;

found = true;

}

}

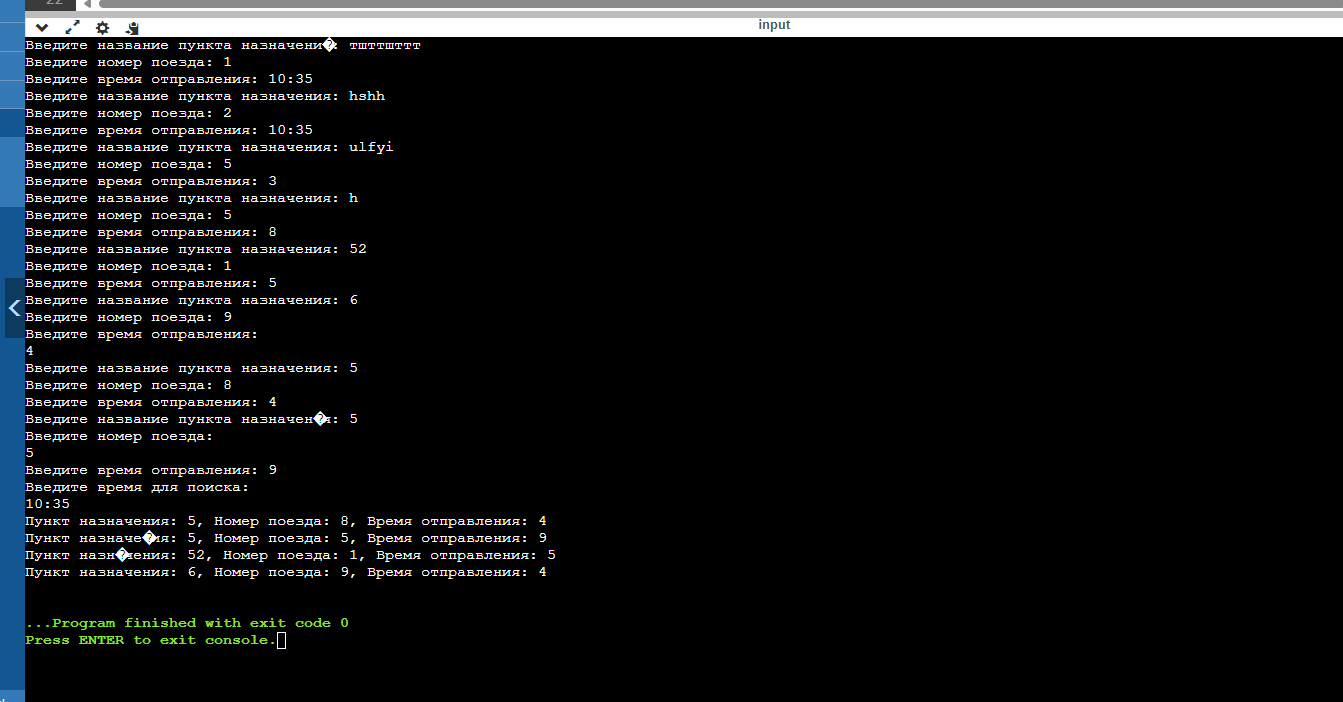
if (!found) {

cout << "Поездов, отправляющихся после указанного времени, нетъ" << endl;

}

return 0;

}



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 <vector>

#include <string>

using namespace std;

struct ExamResult {

string fullName; // ФИО

int numExams;

vector<int> grades; // Оценки

};

// Функция для проверки на 4 и 5

bool allExamsPassed(const ExamResult& result) {

for (int grade : result.grades) {

if (grade < 4) {

return false;

}

}

return true;

}

// Успеваемость

double calculatePerformance(const vector<ExamResult>& results) {

int passedCount = 0;

for (const ExamResult& result : results) {

if (allExamsPassed(result)) {

passedCount++;

}

}

return static\_cast<double>(passedCount) / results.size() \* 100;

}

int main() {

setlocale(0, "");

vector<ExamResult> examResults = { // Пример

{"кто кто", 3, {5, 4, 5}},

{"кто то", 4, {3, 4, 4, 5}},

{"кто то", 2, {4, 5}}

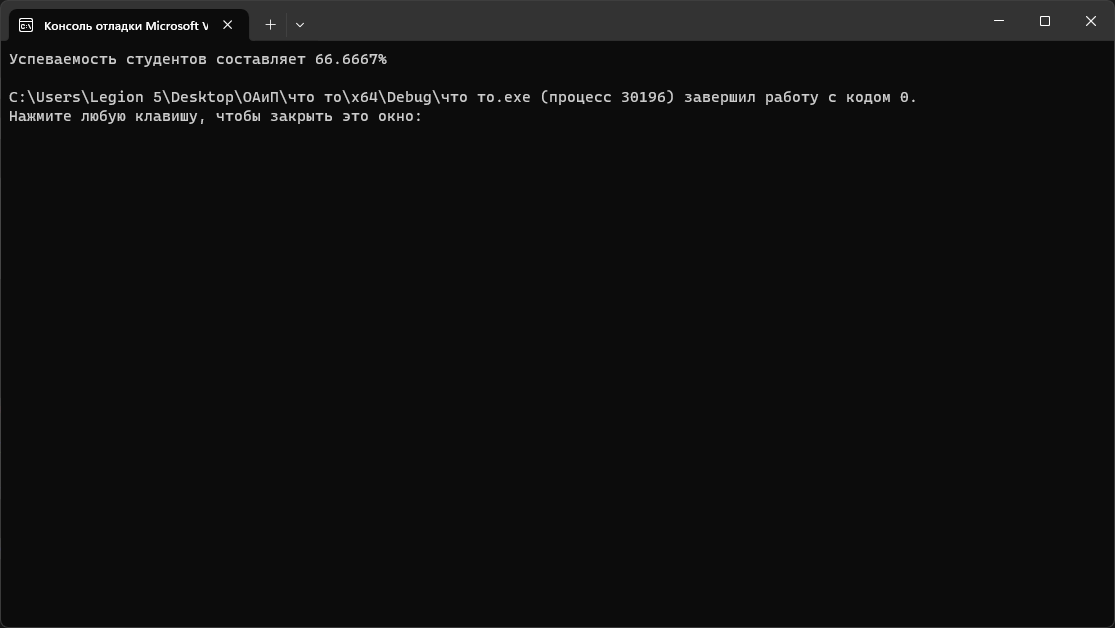
};

double performance = calculatePerformance(examResults);

cout << "Успеваемость студентов составляет " << performance << "%" << endl;

return 0;

}



1. 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.

Это ужас.

