***Министерство образования Республики Беларусь***

***Учреждение Образования***

***«Брестский Государственный Технический Университет»***

***Кафедра ИИТ***

**Лабораторная работа №9**

**По дисциплине ОАиП за II семестр**

**Тема: «Структуры, объединения, перечисления»**

**Выполнил:**

Студент 1-го курса

Группы ИИ-15(1)

Волк И. А.

**Проверила:**

Хацкевич М. В.

Брест 2018

Цель: научиться работать со структурами.

Задание 1.

Создайте структуру (в соответствии с вариантом задания задания), данная структура должна содержать: битовое поле, поле типа - объединение, поле типа - перечисление. Ввести массив структур в соответствии с вариантом. Рассортировать массив в алфавитном порядке по первому полю, входящему в структуру. В программе реализовать меню:

1) Ввод массива структур;

3) Вывод массива структур;

2) Сортировка массива структур;

3) Поиск в массиве структур по заданному параметру;

4) Изменение заданной структуры;

5) Удаление структуры из массива;

6) Вывод на экран массива структур;

7) Выход.

Код программы:

#include "stdafx.h"

#include <Windows.h>

#include <iostream>

#include <stdio.h>

using namespace std;

const int Notname = system("color f0");

/\*\* Структрура, хронящая данные о сотруднике \*\*/

struct sotrudnik

{

unsigned y\_of\_b : 15; // год рождения

bool isSInt : 1; // целочисленная ли зарплата

// объединение для зарплаты

union

{

int s\_int; // для целочисленных зарплат

double s\_double; // для нецелочисленных зарплат

}salary;

// Должность

enum

{

cleaner = 1,

programmer,

director

}position;

char last\_name[100], name[100], patronymic[100]; // фамилия, имя, отчество

int num;

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// выведение сведений о данном сотруднике

void output\_sotr(sotrudnik sotr)

{

cout << "Сотрудник №" << sotr.num << endl << endl;

cout << "Фамилия: " << sotr.last\_name << endl;

cout << "Имя: " << sotr.name << endl;

cout << "Отчество: " << sotr.patronymic << endl;

cout << "Год рождения: " << sotr.y\_of\_b << endl;

cout << "Должность: " << ((sotr.position == 1) ? "Уборщик" : ((sotr.position == 2) ? "Программист" : "Директор")) << endl;

cout << "Зарплата: " << ((sotr.isSInt) ? sotr.salary.s\_int : sotr.salary.s\_double) << endl << endl;

}

int main()

{

setlocale(LC\_ALL, "russian");

sotrudnik\* sotr = new sotrudnik[1]; // указатель для массива структур

int sotr\_num; // количество сотрудников

double isSotr = false; // показывает, существует ли сотрудник

while(1)

{

system("cls");

/\*\* меню \*\*/

int input\_menu;

cout << " 1. Ввод массива сотрудников\n";

cout << " 2. Вывод массива сотрудников\n";

cout << " 3. Сортировка массива сотрудников\n";

cout << " 4. Поиск в массиве сотрудников по заданному параметру\n";

cout << " 5. Изменение заданного сотрудника\n";

cout << " 6. Удаление сотрудника из массива\n";

cout << " 7. Выход\n";

cout << "Ввод: "; cin >> input\_menu;

/\*\*\*\*\*\*\*\*\*\*\*/

system("cls");

// если нужно ввести массив структур

if(input\_menu==1)

{

isSotr = true;

// создание массива с заданным количеством сотрудников

cout << "Введите колическво сотрудников: "; cin >> sotr\_num;

delete[] sotr;

sotr = new sotrudnik[sotr\_num];

// заполнение структуры каждого сотрудника

for (int i = 0; i < sotr\_num; i++)

{

system("cls");

sotr[i].num = i + 1;

cout << "Сотрудник №" << sotr[i].num << endl << endl;

// фамилия, имя, отчество

cout << "Введите фамилию: "; cin >> sotr[i].last\_name; cout << endl;

cout << "Введите имя: "; cin >> sotr[i].name; cout << endl;

cout << "Введите отчество: "; cin >> sotr[i].patronymic; cout << endl;

// год рождения

int y\_of\_b;

cout << "Введите год рождения: "; cin >> y\_of\_b; cout << endl;

sotr[i].y\_of\_b = y\_of\_b;

// должность

int pos;

cout << "Выберите должность (1 - uborshik, 2 - programmist, 3 - direktor): ";

cin >> pos; cout << endl;

if (pos == 1)sotr[i].position = sotrudnik::cleaner;

else if (pos == 2)sotr[i].position = sotrudnik::programmer;

else if (pos == 3)sotr[i].position = sotrudnik::director;

else { system("pause"); return 0; };

// зарплата

double sal;

cout << "Введите зарплату: "; cin >> sal; cout << endl;

if ((sal - (double)((int)sal)) == 0)

{ sotr[i].salary.s\_int = sal; sotr[i].isSInt = true; }

else { sotr[i].salary.s\_double = sal; sotr[i].isSInt = false; }

}

}

// если нужно вывести массив структур

else if (input\_menu == 2)

{

// если сотрудников нет, то можно выполнить только 1-й пункт

if (!isSotr) { cout << "Error!"; Sleep(1000); continue; }

// вывод массива структур

for (int i = 0; i < sotr\_num; i++)

{

output\_sotr(sotr[i]);

}

system("pause");

}

// если нужно сортировать массив структур

else if (input\_menu == 3)

{

// если сотрудников нет

if (!isSotr||(sotr\_num < 2)) { cout << "Error!"; Sleep(1000); continue; }

int type\_sort; // тип сортировки

cout << " 1. По фамилии \n 2. По году рождения \n 3. По должности\nВвод: ";

cin >> type\_sort;

sotrudnik sotr\_copy; //копия сотрудника

for (int i = 0; i < (sotr\_num-1); i++)

{

// сортировка по фамилии

if (type\_sort == 1)

{

if(sotr[i].last\_name[0]>sotr[i+1].last\_name[0])

{

int sotr\_i\_num = sotr[i].num;

sotr[i].num = sotr[i + 1].num;

sotr[i + 1].num = sotr\_i\_num;

sotr\_copy = sotr[i];

sotr[i] = sotr[i + 1];

sotr[i + 1] = sotr\_copy;

}

}

// сортировка по году рождения

else if (type\_sort == 2)

{

if (sotr[i].y\_of\_b > sotr[i + 1].y\_of\_b)

{

int sotr\_i\_num = sotr[i].num;

sotr[i].num = sotr[i + 1].num;

sotr[i + 1].num = sotr\_i\_num;

sotr\_copy = sotr[i];

sotr[i] = sotr[i + 1];

sotr[i + 1] = sotr\_copy;

}

}

// сортировка по должности

else if (type\_sort == 3)

{

if (sotr[i].position > sotr[i + 1].position)

{

int sotr\_i\_num = sotr[i].num;

sotr[i].num = sotr[i + 1].num;

sotr[i + 1].num = sotr\_i\_num;

sotr\_copy = sotr[i];

sotr[i] = sotr[i + 1];

sotr[i + 1] = sotr\_copy;

}

}

else { cout << "Error sorting!"; continue; }

}

system("cls");

cout << "Success!";

Sleep(1000);

}

// если нужно производить поиск в массиве по заданному параметру

else if (input\_menu == 4)

{

if (!isSotr) { cout << "Error!"; Sleep(1000); continue; } // если сотрудников нет

int type\_search; // параметр поиска

cout << " 1. По фамилии\n 2. По имени\n 3. По отчеству\n 4. По году рождения\n 5. По должности\n 6. По зарплате\nВвод: ";

cin >> type\_search;

system("cls");

// поиск по фамилии

if (type\_search == 1)

{

char last\_name[100]=""; cout << "Ввод: "; cin >> last\_name;

system("cls");

for (int i = 0; i < sotr\_num; i++)

if (!strcmp(last\_name, sotr[i].last\_name))

output\_sotr(sotr[i]);

system("pause");

}

// поиск по имени

else if (type\_search == 2)

{

char name[100]=""; cout << "Ввод: "; cin >> name;

system("cls");

for (int i = 0; i < sotr\_num; i++)

if (!strcmp(name, sotr[i].name))

output\_sotr(sotr[i]);

system("pause");

}

// поиск по отчеству

else if (type\_search == 3)

{

char patronymic[100]=""; cout << "Ввод: "; cin >> patronymic;

system("cls");

for (int i = 0; i < sotr\_num; i++)

if (!strcmp(patronymic, sotr[i].patronymic))

output\_sotr(sotr[i]);

system("pause");

}

// поиск по году рождения

else if (type\_search == 4)

{

int y\_of\_b; cout << "Ввод: "; cin >> y\_of\_b;

system("cls");

for (int i = 0; i < sotr\_num; i++)

if(y\_of\_b==sotr[i].y\_of\_b)

output\_sotr(sotr[i]);

system("pause");

}

// поиск по должности

else if (type\_search == 5)

{

int position; cout << "Ввод: "; cin >> position;

system("cls");

for (int i = 0; i < sotr\_num; i++)

if (position == sotr[i].position)

output\_sotr(sotr[i]);

system("pause");

}

// поиск по зарплате

else if (type\_search == 6)

{

bool isSInt = false;

double salary; cout << "Ввод: "; cin >> salary;

if ((salary - (double)((int)salary)) == 0.0) isSInt = true;

system("cls");

for (int i = 0; i < sotr\_num; i++)

if (((isSInt)?(int)salary:salary) == ((sotr[i].isSInt)?sotr[i].salary.s\_int:sotr[i].salary.s\_double))

output\_sotr(sotr[i]);

system("pause");

}

else { cout << "Error searching!"; continue; }

system("cls");

cout << "Success!";

Sleep(1000);

}

// если нужно изменить заданную структуру

else if (input\_menu == 5)

{

if (!isSotr) { cout << "Error!"; Sleep(1000); continue; } // если сотрудников нет

unsigned short num; // номер сотрудника

cout << "Номер сотрудника: "; cin >> num;

unsigned short i = num - 1; // номер сотрудника - 1 (для работы с массивом)

// если неправильно введен номер сотрудника

if (num > sotr\_num || num == 0) { cout << "Error!"; continue; }

system("cls");

cout << "Сотрудник №" << sotr[i].num << endl << endl;

// фамилия, имя, отчество

cout << "Введите фамилию: "; cin >> sotr[i].last\_name; cout << endl;

cout << "Введите имя: "; cin >> sotr[i].name; cout << endl;

cout << "Введите отчество: "; cin >> sotr[i].patronymic; cout << endl;

// год рождения

int y\_of\_b;

cout << "Введите год рождения: "; cin >> y\_of\_b; cout << endl;

sotr[i].y\_of\_b = y\_of\_b;

// должность

int pos;

cout << "Выберите должность (1 - uborshik, 2 - programmist, 3 - direktor): ";

cin >> pos; cout << endl;

if (pos == 1)sotr[i].position = sotrudnik::cleaner;

else if (pos == 2)sotr[i].position = sotrudnik::programmer;

else if (pos == 3)sotr[i].position = sotrudnik::director;

else { system("pause"); return 0; };

// зарплата

double sal;

cout << "Введите зарплату: "; cin >> sal; cout << endl;

if ((sal - (double)((int)sal)) == 0)

{ sotr[i].salary.s\_int = sal; sotr[i].isSInt = true; }

else { sotr[i].salary.s\_double = sal; sotr[i].isSInt = false; }

system("cls");

cout << "Success!";

Sleep(1000);

}

// если нужно удалить структурту из массива

else if (input\_menu == 6)

{

if (!isSotr) { cout << "Error!"; Sleep(1000); continue; } // если сотрудников нет

unsigned short num; // номер сотрудника

cout << "Номер сотрудника: "; cin >> num;

// если неправильно введен номер сотрудника

if (num > sotr\_num || num == 0) { cout << "Error!"; continue; }

// если в массиве всего один элемент

if (sotr\_num == 1)

{

delete[] sotr;

sotr = new sotrudnik[1];

isSotr = false;

system("cls");

cout << "Success!";

Sleep(1000);

continue;

}

sotr\_num = sotr\_num - 1;

// сдвиг

for (unsigned short i = num - 1; i < sotr\_num; i++)

{

int sotr\_i\_num = sotr[i].num;

sotr[i].num = sotr[i + 1].num;

sotr[i + 1].num = sotr\_i\_num;

sotr[i] = sotr[i + 1];

}

// создем копию, в которую размещаем нужный массив

sotrudnik \* sotr\_copy = new sotrudnik[sotr\_num];

for (int i = 0; i < sotr\_num; i++)

{

sotr\_copy[i] = sotr[i];

}

// изменяем существующий массив сотрудников

delete[] sotr;

sotr = new sotrudnik[sotr\_num];

for (int i = 0; i < sotr\_num; i++)

{

sotr[i] = sotr\_copy[i];

}

system("cls");

cout << "Success!";

Sleep(1000);

}

// выход

else if (input\_menu == 7)

break;

// если не выбран ни один пункт меню

else

{

cout << " Error! ";

Sleep(500);

system("cls");

}

}

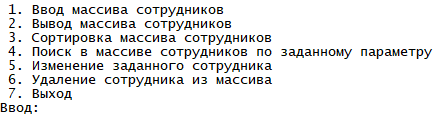
system("pause");

return 0;

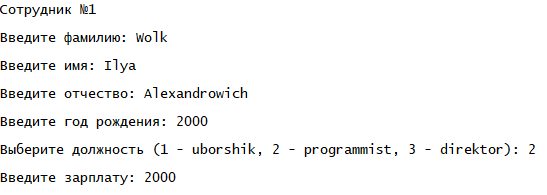
}

Вывод:

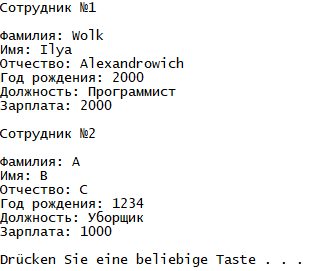
Меню:



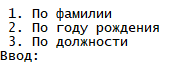
1) Ввод



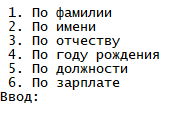
2) Вывод



3) Сортировка



4) Поиск



5) Изменение



6) Удаление



Вывод: по ходу лабораторной работы научился работать со структурами.

Блок схема пункта №3 (сортировка) меню:

