***Министерство образования Республики Беларусь***

***Учреждение Образования***

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

***Кафедра ИИТ***

**Лабораторная работа №1**

**По дисциплине ОАиП за II семестр**

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

**Выполнил:**

Студент 1-го курса

Группы ИИ-18

Смоляков А.Д.

**Проверил:**

Михно Е. В.

Брест 2020

**Лабораторная работа №1**

Цель: научиться работать со структурами.

**Вариант 5**

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 <Windows.h>

#include <iostream>

#include <stdio.h>

using namespace std;

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

struct reader {

char last\_name[100], name[100], patronymic[100];

int number;

enum {

TheHoundoftheBaskervilles = 1,

GameofThrones,

TheChroniclesofNarnia,

HarryPotter

}position;

unsigned y\_of\_d;

int num;

};

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

// выведение сведений о данном читателе

void output\_read(reader read) {

cout << "Читатель №" << read.num << endl << endl;

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

cout << "Имя: " << read.name << endl;

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

cout << "Читательский билет № " << read.number << endl;

cout << "Название книги: " << ((read.position == 1) ? "Шерлок Холмс. Собака Баскервиль" : ((read.position == 2) ? "Игра Престолов" : ((read.position == 3) ? "Хроники Нарнии" : "Гарри Поттер и Узник Азкабана"))) << endl;

cout << "Срок возврата: " << read.y\_of\_d << endl << endl;

}

void sort(reader\* read, int read\_num)

{

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

{

if (read[i].num > read[i + 1].num)

{

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

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

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

reader temp = read[i];

read[i] = read[i + 1];

read[i + 1] = temp;

}

}

}

void find(reader\* read, int read\_num)

{

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

system("cls");

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

if (num == read[i].num)

output\_read(read[i]);

system("pause");

}

int main() {

int a;

setlocale(LC\_ALL, "russian");

reader\* read = new reader[1]; // указатель для массива структур

int read\_num; // количество читателей

double isRead = false; // показывает, существует ли читатель

while (1) {

system("cls");

/\*\* меню \*\*/

int input\_menu;

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

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

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

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

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

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

cout << " 7. Выход\n";

cout << "Ввод: "; cin >> input\_menu;

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

system("cls");

if (input\_menu == 1) {

isRead = true;

// создание массива с заданным количеством читателей

cout << "Введите колическво читателей: "; cin >> read\_num;

delete[] read;

read = new reader[read\_num];

// заполнение структуры каждого читателя

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

system("cls");

read[i].num = i + 1;

cout << "Читатель №" << read[i].num << endl << endl;

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

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

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

cout << "Читательский билет № "; cin >> read[i].number; cout << endl;

cout << "Выберите книгу (1 - The Hound of the Baskervilles, 2 - Game of Thrones, 3 - The Chronicles of Narnia, 4 - Harry Potter): ";

int pos;

cin >> pos; cout << endl;

if (pos == 1)read[i].position = reader::TheHoundoftheBaskervilles;

else if (pos == 2)read[i].position = reader::GameofThrones;

else if (pos == 3)read[i].position = reader::TheChroniclesofNarnia;

else if (pos == 4)read[i].position = reader::HarryPotter;

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

int y\_of\_d;

cout << "Срок возврата: "; cin >> y\_of\_d; cout << endl;

read[i].y\_of\_d = y\_of\_d;

}

}

else if (input\_menu == 2) {

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

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

// вывод массива структур

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

output\_read(read[i]);

}

system("pause");

}

// если нужно сортировать массив структур

else if (input\_menu == 3)

{

// если читателей нет

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

int type\_read;

cout << " 1. По фамилии \n 2. По номеру читательского билета \n 3. По выбранной книге\nВвод: ";

cin >> type\_read;

reader read\_copy;

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

{

// сортировка по фамилии

if (type\_read == 1)

{

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

{

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

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

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

read\_copy = read[i];

read[i] = read[i + 1];

read[i + 1] = read\_copy;

}

system("cls");

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

output\_read(read[i]);

}

system("pause");

}

// сортировка по номеру читательского билета

else if (type\_read == 2)

{

void sort();

system("cls");

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

output\_read(read[i]);

}

system("pause");

}

// сортировка по выбранной книге

else if (type\_read == 3)

{

void sort();

system("cls");

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

output\_read(read[i]);

}

system("pause");

}

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

}

system("cls");

}

// если нужно производить поиск в массиве по заданному параметру

else if (input\_menu == 4)

{

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

int type\_search; // параметр поиска

cout << " 1. По фамилии\n 2. По имени\n 3. По отчеству\n 4. По номеру читательского билета\n 5. По названию\n 6. По сроку возврата\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 < read\_num; i++)

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

output\_read(read[i]);

system("pause");

}

else if (1 < type\_search < 7)

{

void find();

}

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

system("cls");

Sleep(1000);

}

// если нужно изменить заданную структуру

else if (input\_menu == 5)

{

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

unsigned short num; // номер читателя

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

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

// если неправильно введен номер читателя

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

system("cls");

cout << "Читатель №" << read[i].num << endl << endl;

// фамилия, имя, отчество

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

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

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

// номер читательского билета

int number;

cout << "Введите номер читательского билета: "; cin >> number; cout << endl;

read[i].number = number;

// книга

int pos;

cout << "Выберите книгу (1 - The Hound of the Baskervilles, 2 - Game of Thrones, 3 - The Chronicles of Narnia, 4 - Harry Potter): ";

cin >> pos; cout << endl;

if (pos == 1)read[i].position = reader::TheHoundoftheBaskervilles;

else if (pos == 2)read[i].position = reader::GameofThrones;

else if (pos == 3)read[i].position = reader::TheChroniclesofNarnia;

else if (pos == 4)read[i].position = reader::HarryPotter;

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

// срок возврата

int y\_of\_d;

cout << "Введите срок возврата: "; cin >> y\_of\_d; cout << endl;

read[i].y\_of\_d = y\_of\_d;

system("cls");

cout << "Хотите вывести массив(1-Да, 0-Нет): "; cin >> a;

system("cls");

if (a == 1) {

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

output\_read(read[i]);

}

system("pause");

}

else if (a == 0)

system("pause");

}

// если нужно удалить структурту из массива

else if (input\_menu == 6)

{

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

unsigned short num; // номер читателя

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

// если неправильно введен номер читателя

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

// если в массиве всего один элемент

if (read\_num == 1)

{

delete[] read;

read = new reader[1];

isRead = false;

system("cls");

Sleep(1000);

continue;

}

read\_num = read\_num - 1;

// сдвиг

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

{

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

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

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

read[i] = read[i + 1];

}

// создем копию, в которую размещаем нужный массив

reader \* read\_copy = new reader[read\_num];

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

{

read\_copy[i] = read[i];

}

// изменяем существующий массив читателей

delete[] read;

read = new reader[read\_num];

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

{

read[i] = read\_copy[i];

}

cout << "Хотите вывести массив(1-Да, 0-Нет): "; cin >> a;

system("cls");

if (a == 1) {

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

output\_read(read[i]);

}

system("pause");

}

else if (a == 0)

system("pause");

}

// выход

else if (input\_menu == 7)

break;

// если не выбран ни один пункт меню

else

{

cout << " Error! ";

Sleep(500);

system("cls");

}

}

system("pause");

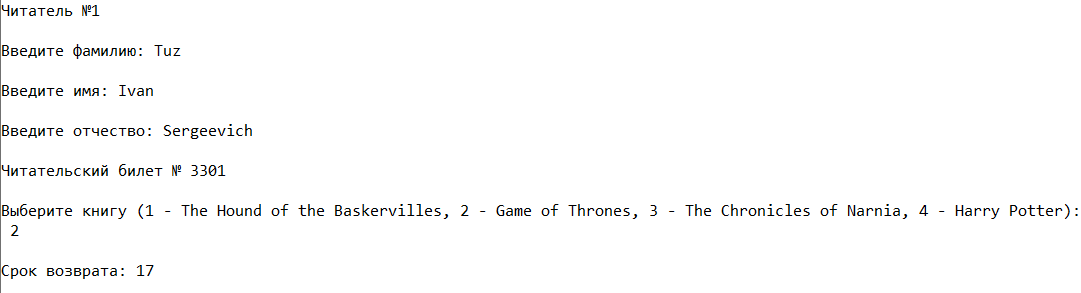
return 0;

}

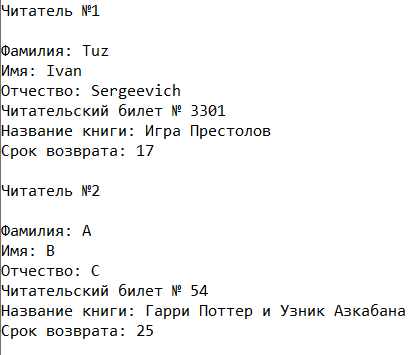
Вывод:

Меню:

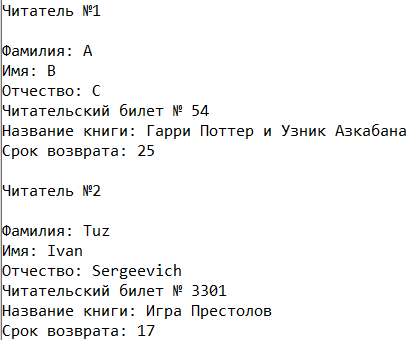
1) Ввод



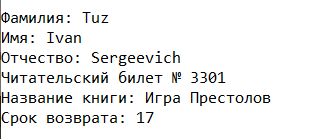
2) Вывод



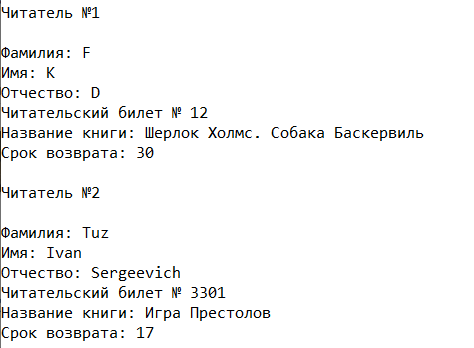
3) Сортировка



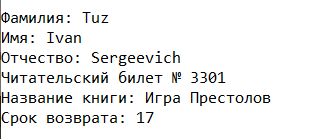
4) Поиск



5) Изменение



6) Удаление



Вывод: по ходу лабораторной работы научился работать со структурами.