Отчёт по практике

**Выполнил: Гоков Руслан   
Группа: WEB-012**

**Chișinău 2021**

**Содержание:**

1. **Алгоритм (3)**
2. **Код программы (4-6)**
3. **Тестирование программы (7)**
4. **Выводы (8)**
5. **Библиография (9)**
6. **Приложения (10)**

**1. Алгоритм**

**1. Открываем необходимые библиотек: fstream, string,iostream.  
2. Объявляем массив структур.  
 2.1) В структуре объявляем необходимые переменные типов string и int.  
 2.2) Объявляем переменную типа int для хранения в ней кол-ва переменных внутри структуры.  
3. Создаём подпрограмму для чтения данных из файла.  
 3.1) Открываем файл для чтения с помощью функции ifstream.  
 3.2) Объявляем цикл while который вносит данные из файла в ячейки массива структур.  
 3.3) Присваиваем переменной NumBooks максимальное значение i, – 1, потому что отчёт начинается с 0.  
4. Создаём подпрограмму которая будет выводить данные из структуры.  
5. Создаём подпрограмму для перезаписи файла, она пригодится для удаления выбранной книги.  
6. Создаём подпрограмму для регистрации новой книги.  
6.1) Открываем файл для чтения и записи, и используем функция позволяющую записывать новые данные файлы, без потери предыдущих данных.  
6.2) NumBooks++ создаст ячейку в массиве структур для новой книги  
6.3) Присваиваем элементу массива последнее место для того чтобы новая книга была в конце списка.  
6.4) Переносим данные в файл   
7. Создаём подпрограмму для поиска позиции жанра книги в строке, которая так же пригодиться для удаления книги.  
8. Создаём подпрограмму для удаления книги по жанру.  
8.1) Объявляем временную переменную типа string, и предлагаем пользователю определить её.  
8.2) Присваиваем переменной pos найденную подпрограммой позицию временной переменной  
8.3) Если жанр не найден выводим соответствующее предложение.  
8.4) В противном случае запускаем цикл который заменить следующую строку на место удалённой  
8.5) Вызываем подпрограмму перезаписи файла, которая запишет в файл данные без удалённой книги.  
10. Создаём подпрограмму для поиска наибольшего кол-ва страниц в книге.  
10.1) Присваиваем переменной max кол-во страниц в первой книге.**

**10.2) С помощью цикла сравниваем кол-ва страниц во всех книгах.  
10.3) Выводим максимальное найденное кол-во страниц.  
11. Создаём подпрограмму для поиска цены иностранных книг.  
11.1) Объявляем переменные для поиска суммы.  
11.2) Объявляем переменные для поиска иностранных книг   
11.3) С помощью цикла ищем ячейки с интересующими нас языками.  
11.4) Считаем сумму и делим на кол-во выполненных действий цикла.**

**Код.**

#include <iostream>

#include <fstream>

using namespace std;

struct RegBook {

int CodCarte;

string Num\_Prenum\_Autor;

string Titlu;

int AnEd;

string Limba;

float Pret;

int Npag;

string Tema;

}Books[50];

int NumBooks;

//Регестрация книги

void ReadFile()

{

ifstream File;

int i = 0;

File.open("C:\\Users\\User\\Desktop\\Практика\\Программирование\_два\\Carti\_two.txt");

while (!File.eof())

{

File >> Books[i].CodCarte >> Books[i].Num\_Prenum\_Autor >> Books[i].Titlu >> Books[i].AnEd >> Books[i].Limba >> Books[i].Pret >> Books[i].Npag >> Books[i].Tema;

i++;

}

NumBooks = i - 1;

File.close();

}

void CoutBook()

{

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

{

cout << Books[i].CodCarte << " " << Books[i].Num\_Prenum\_Autor << " " << Books[i].Titlu << " " << Books[i].AnEd << " " << Books[i].Limba << " " << Books[i].Pret << " " << Books[i].Npag << " " << Books[i].Tema << endl;

}

}

void AppendBook()

{

fstream File;

File.open("C:\\Users\\User\\Desktop\\Практика\\Программирование\_два\\Carti\_two.txt", ios::app);

NumBooks++;

cout << "Введите код книги:";

cin >> Books[NumBooks - 1].CodCarte;

cout << "Введите Имя\_Фамилию\_Автра через нижю черту:";

cin >> Books[NumBooks - 1].Num\_Prenum\_Autor;

cout << "Введите Название книги:";

cin >> Books[NumBooks - 1].Titlu;

cout << "Введите год выпуска книги:";

cin >> Books[NumBooks - 1].AnEd;

cout << "Введите Язык книги (Rum,En,Ru):";

cin >> Books[NumBooks - 1].Limba;

cout << "Введите цену книги:";

cin >> Books[NumBooks - 1].Pret;

cout << "Введите количество страниц книги:";

cin >> Books[NumBooks - 1].Npag;

cout << "Введите тему книги:";

cin >> Books[NumBooks - 1].Tema;

File << Books[NumBooks - 1].CodCarte << " " << Books[NumBooks - 1].Num\_Prenum\_Autor << " " << Books[NumBooks - 1].Titlu << " " << Books[NumBooks - 1].AnEd << " " << Books[NumBooks - 1].Limba << " " << Books[NumBooks - 1].Pret << " " << Books[NumBooks - 1].Npag << " " << Books[NumBooks - 1].Tema << endl;

File.close();

}

//Удаление книги с указаным жанром

void refile()

{

ofstream file("C:\\Users\\User\\Desktop\\Практика\\Программирование\_два\\Carti\_two.txt");

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

file << Books[i].CodCarte << " " << Books[i].Num\_Prenum\_Autor << " " << Books[i].Titlu << " " << Books[i].AnEd << " " << Books[i].Limba << " " << Books[i].Pret << " " << Books[i].Npag << " " << Books[i].Tema << endl;

file.close();

}

int findpos(string str)

{

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

if (Books[i].Tema == str) return i;

return -1;

}

void deletebook()

{

string temp;

cout << "Введите жанр для удаления :";

cin >> temp;

int pos = findpos(temp);

if (pos == -1)cout << "Жанр не найден" << endl;

else

{

for (int i = pos; i < NumBooks - 1; i++)

Books[i] = Books[i + 1];

NumBooks--;

refile();

}

}

// Книга по году

void ageg()

{

int age;

cout << "Сортировка по году. \n";

cout << "Введите нужный вам год : ";

cin >> age;

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

{

if (age == Books[i].AnEd)

{

cout << Books[i].CodCarte << " " << Books[i].Num\_Prenum\_Autor << " " << Books[i].Titlu << " " << Books[i].AnEd << " " << Books[i].Limba << " " << Books[i].Pret << " " << Books[i].Npag << " " << Books[i].Tema << endl;

}

}

}

//Книги на румынском и сортировка

void Rumm()

{

cout << "Введите название книги : ";

string Name;

cin >> Name;

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

{

if (Name == Books[i].Titlu)

{

if (Books[i].Limba == "Rum")

{

fstream FileRum;

FileRum.open("C:\\Users\\User\\Desktop\\Практика\\Программирование\_два\\Romana.txt", ios::app);

FileRum << Books[i].CodCarte << " " << Books[i].Num\_Prenum\_Autor << " " << Books[i].Titlu << " " << Books[i].AnEd << " " << Books[i].Limba << " " << Books[i].Pret << " " << Books[i].Npag << " " << Books[i].Tema << endl;

FileRum.close();

}

}

}

}

void Sortirovka()

{

string temp;

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

for (int j = i + 1; j < NumBooks; j++)

if (Books[i].Num\_Prenum\_Autor < Books[j].Num\_Prenum\_Autor)

{

temp = Books[i].Num\_Prenum\_Autor;

Books[i] = Books[j];

Books[j].Num\_Prenum\_Autor = temp;

}

}

//Максимальное количество страниц

void Npag()

{

int max = Books[0].Npag;

for (int i = 1; i < NumBooks; i++)

{

if (Books[i].Npag > max) max = Books[i].Npag;

}

cout << "Максимальное количество страниц - " << max << endl;

}

//Средняя ценя книги на иностранном языке

void sredEn()

{

ReadFile();

float DopVariable = 0;

float d = 0;

float sredPret;

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

{

if (Books[i].Limba == "En")

{

//if (Books[i].Pret > 0)

//{

DopVariable = DopVariable + Books[i].Pret;

d++;

//}

}

}

sredPret = DopVariable / d;

std::cout << "Средняя цена книги на иностранном языке : ";

printf("%.2f", sredPret);

}

//Дополнительное задание

int dop()

{

int h;

int dopp;

cout << "Введите ! числа - !";

cin >>h;

dopp = 1;

for (int i = 1; i <= h; i++)

{

dopp = dopp \* i;

}

cout << "\n!" << h << "=" << dopp;

return 0;

}

//Повтор

int povtor(int povtoren)

{

cout << "\nХотите перейти в меню ?\n";

cout << "1.Да\n";

cout << "2.Нет\n";

cin >> povtoren;

return 0;

}

int main()

{

setlocale(LC\_ALL, "ru");

int n;

int povtoren=0 ;

while (povtoren != 2 )

{

cout << "Библиотека.\n";

cout << "1.Регистрация книги.\n";

cout << "2.Книги в библиотеке.\n";

cout << "3.Удаление книг по указоному жанру.\n";

cout << "4.Поиск книги по году.\n";

cout << "5.Перенос книги на румынском.\n";

cout << "6.Максимальное количество страниц.\n";

cout << "7.Средняя ценя книги на иностранном языке.\n";

cout << "8.Дополнительное задание.\n";

cout << "9.Выход.\n";

cout << "Пунтк : ";

cin >> n;

cout << endl;

system("cls");

switch (n)

{

case 1:

AppendBook();

povtor(povtoren);

cout << "\nХотите перейти в меню ?\n";

cout << "1.Да\n";

cout << "2.Нет\n";

system("cls");

break;

case 2:

ReadFile();

CoutBook();

//povtor(povtoren);

cout << "\nХотите перейти в меню ?\n";

cout << "1.Да\n";

cout << "2.Нет\n";

cin >> povtoren;

system("cls");

break;

case 3:

ReadFile();

CoutBook();

cout << endl;

deletebook();

cout << endl;

system("cls");

ReadFile();

CoutBook();

cout << "\nХотите перейти в меню ?\n";

cout << "1.Да\n";

cout << "2.Нет\n";

cin >> povtoren;

system("cls");

break;

case 4 :

ReadFile();

ageg();

cout << "\nХотите перейти в меню ?\n";

cout << "1.Да\n";

cout << "2.Нет\n";

cin >> povtoren;

system("cls");

break;

case 5: ReadFile();

CoutBook();

Sortirovka();

Rumm();

cout << "\nХотите перейти в меню ?\n";

cout << "1.Да\n";

cout << "2.Нет\n";

cin >> povtoren;

system("cls");

break;

case 6:

ReadFile();

Npag();

cout << "\nХотите перейти в меню ?\n";

cout << "1.Да\n";

cout << "2.Нет\n";

cin >> povtoren;

system("cls");

break;

case 7:

ReadFile();

sredEn();

cout << "\nХотите перейти в меню ?\n";

cout << "1.Да\n";

cout << "2.Нет\n";

cin >> povtoren;

system("cls");

break;

case 8:

dop();

cout << "\nХотите перейти в меню ?\n";

cout << "1.Да\n";

cout << "2.Нет\n";

cin >> povtoren;

system("cls");

break;

case 9:cout << "\nТочно хотите выйти ?\n";

cout << "1.Нет\n";

cout << "2.Да\n";

cin >> povtoren;

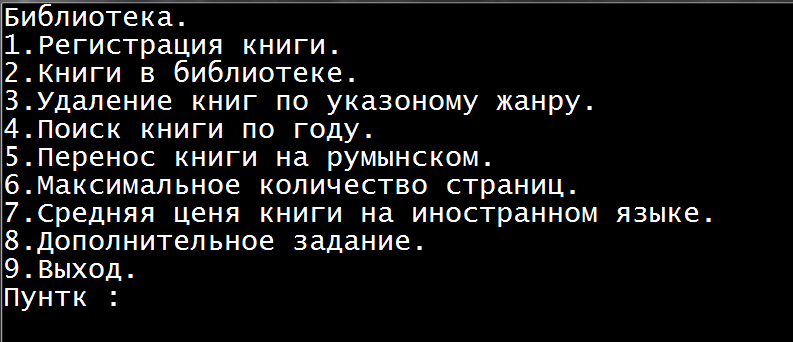
break;

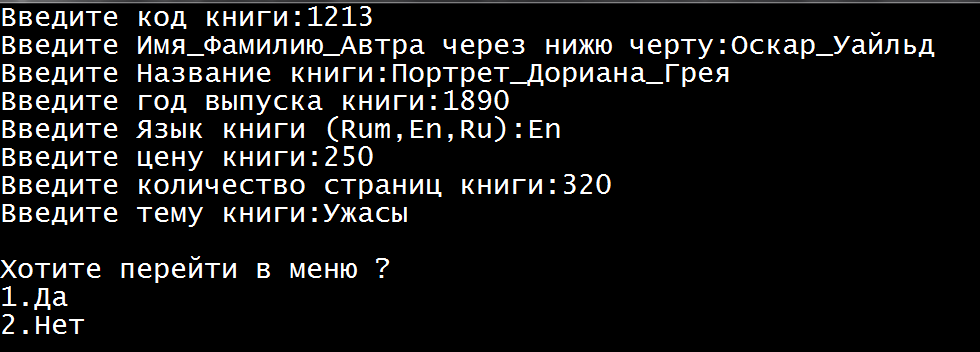
}

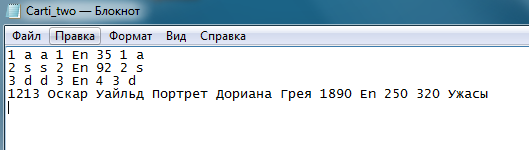
}

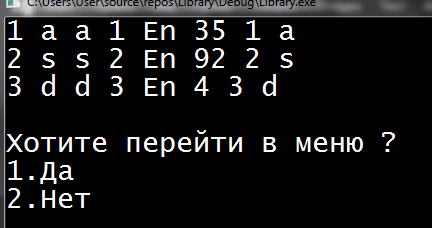
}

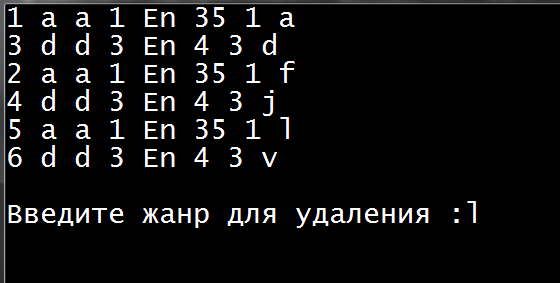
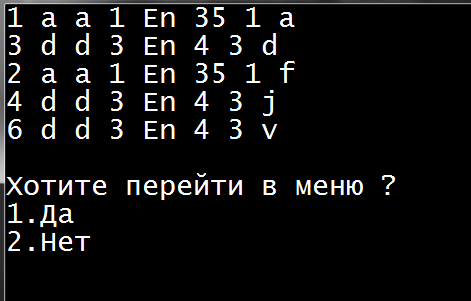
**Результат тестирования подпрограмм и табличного процессора:**

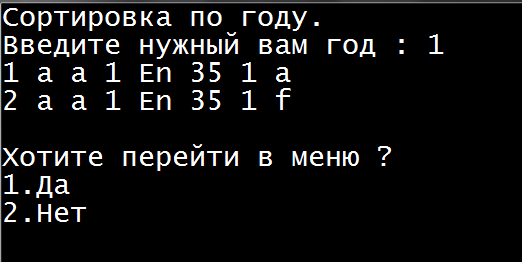
**1) Меню:**

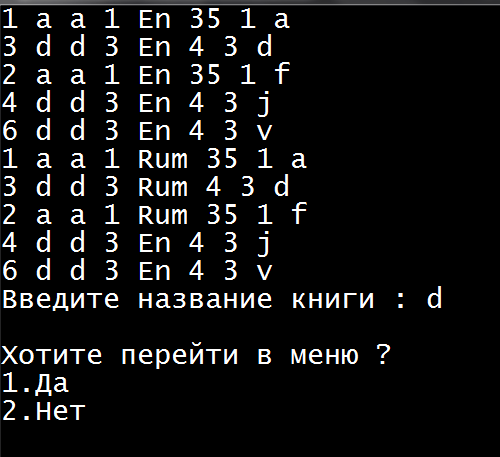
**2) Пункт – 1:**

****

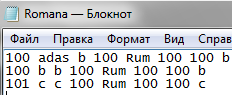
**3) Пункт-2:**

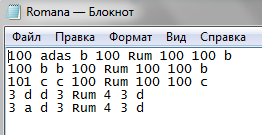
**4) Пункт-3-до: и после:**

**5) Пункт-4:**

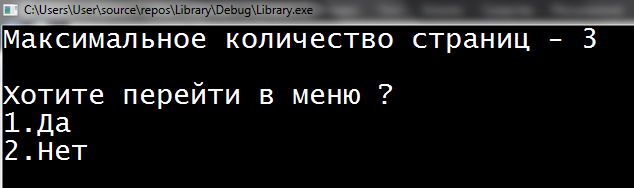
****

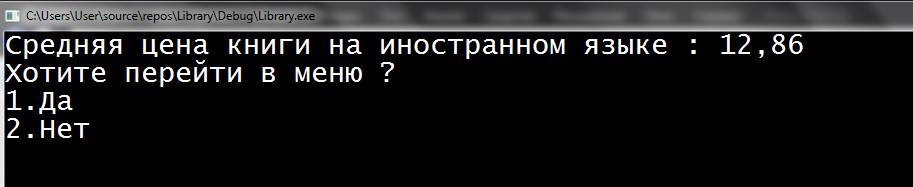
**До:**

**6) Пункт-5:**

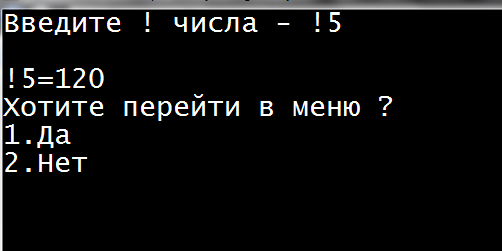


**После:**

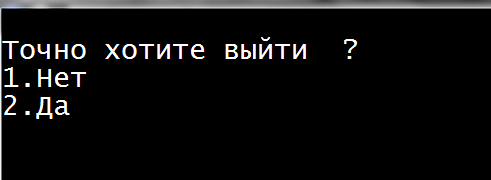
**7) Пункт-6:**

****

**8) Пункт-7:**

****

**9)Пункт-8:**

****

**10)Выход:**

**Вывод**

**-В данной практической работе, я тщательно исследовал и работал с файлами (изучение ввода, вывода и фильтрации информации) и структурами (заполнение массива структур, извлечение информации и сортировка информации). Результаты исследования: Полное выполнение практики за короткий срок.**

**-В данной практической работе сложности сложились только с сортировкой в алфавитном порядке. Эту сложность решила документация найденная в интернете:** [**https://goo-gl.su/1fD6H7**](https://goo-gl.su/1fD6H7) **.**

**-В последующих своих работах, пункты в которых я допускал ошибки будут учитываться и исправляться.**

**-Собственный комментарий: “В этой практической работе, я практически не заметил помех для её реализации, наоборот набрался опыта и новых знаний.” .**

**Библиография.**

**Конспекты:** [**https://classroom.google.com/u/4/w/MjU2ODgyMjQ0NzM0/t/all**](https://classroom.google.com/u/4/w/MjU2ODgyMjQ0NzM0/t/all)

**Сайты с документацией: 1)**[**https://docs.microsoft.com/en-us/cpp/?view=msvc-160**](https://docs.microsoft.com/en-us/cpp/?view=msvc-160)

**2)** [**https://ru.stackoverflow.com/**](https://ru.stackoverflow.com/)

**3)** [**https://coderoad.ru/**](https://coderoad.ru/)