

# ОТЧЕТ по лаб. раб. N1

«Трекер книг»

**ФИО: Чернухин Александр Андреевич**

**Группа: СКБ252**

## **1. Задание**

1. Добавлять информацию о книге:  
Название (до 100 символов),  
Автор (до 50 символов),  
Год прочтения,  
Оценка по 10-балльной шкале (float).
2. Сохранять данные в файл books.txt.
3. Загружать данные из файла при запуске.
4. Показывать список всех книг.
5. Показывать статистику:  
Количество прочитанных книг;  
Средняя оценка;  
Самая высоко оценённая книга.

## **2. Алгоритм работы программы**

Функция загрузки файла

- Проверка существования файла
- Цикл загружает в файл данные о книге
- Заккрытие файла

Функция добавления книги

- Проверка лимита книг
- Поэтапный вывод текста
- Поэтапный ввод данных
- Увеличение счетчика

Функция вывода списка книг

- Проверка наличия книг

- Создание заголовка таблицы
- Цикл, выводящий данные из массива в терминал

Функция вывода статистики

- Проверка наличия книг
- Создание переменных для расчётов
- Цикл, проходящийся по массиву книг и вычисляющий значение средней оценки и максимальной оценки.
- Вывод текста статистики и соответствующих данных

Функция вывода меню

- Вывод текста меню

Функция main

- Вызов функции загрузки файла
- Цикл Do while с использованием ветвления switch case для выбора пункта меню и вызова соответствующей функции

### **3. Код программы**

mymain.cpp × GUIDE.cpp Books.txt

practice\_Book\_Tracker > mymain.cpp > ShowStat()

```
1  #include <iostream>
2  #include <fstream>
3  #include <cstring>
4  #include <iomanip>
5  #include <algorithm>
6  #include <string>
7
8  using namespace std;
9
10 const int MAX_LENGTH_NAME = 100;
11 const int MAX_LENGTH_AUTHOR = 50;
12 const int MAX_NUM_BOOKS = 100;
13
14 struct Book
15 {
16     char name[MAX_LENGTH_NAME];
17     char author[MAX_LENGTH_AUTHOR];
18     char data[8];
19     float score;
20 };
21
22 Book books[MAX_NUM_BOOKS];
23 int cnt = 0;
24
25 void SaveFile(const char *filename, Book NewBook){
26     ofstream fout(filename, ios::app);
27     if (!fout){
28         cout << "Невозможно сохранить файл" << endl;
29         return;
30     }
31
32     string name = NewBook.name;
33     string author = NewBook.author;
34     replace(name.begin(), name.end(), ' ', '_');
35     replace(author.begin(), author.end(), ' ', '_');
36
37     fout << name << " " << author << " " << NewBook.data << " " << NewBook.score << endl;
38     fout.close();
39 }
40
41 void LoadFile(const char *filename){
42     ifstream fin(filename);
43     if (!fin){
44         return;
45     }
46
47     // ...
```

Ln 133, 0

```
mymain.cpp x GUIDE.cpp Books.txt
practice_Book_Tracker > mymain.cpp > ShowStat()
41 void LoadFile(const char *filename){
42     if (!fin){
43
44     string name, author;
45     while (fin >> name >> author >> books[cnt].data >> books[cnt].score) {
46         replace(name.begin(), name.end(), '_', ' ');
47         replace(author.begin(), author.end(), '_', ' ');
48
49         strcpy(books[cnt].name, name.c_str());
50         strcpy(books[cnt].author, author.c_str());
51
52         cnt++;
53     }
54     fin.close();
55 }
56
57 void AddBook(){
58     if (cnt >= MAX_NUM_BOOKS){
59         cout << "Введено максимальное кол-во книг" << endl;
60         return;
61     }
62     Book NewBook;
63
64     cin.ignore();
65
66     cout << "Введите название книги: ";
67     cin.getline(NewBook.name, MAX_LENGTH_NAME);
68
69     cout << "Введите имя автора: ";
70     cin.getline(NewBook.author, MAX_LENGTH_AUTHOR);
71
72     cout << "Введите дату прочтения (дд.мм.гг): ";
73     cin >> NewBook.data;
74
75     cout << "Введите оценку (0-10): ";
76     cin >> NewBook.score;
77
78     if((NewBook.score < 0) || (NewBook.score > 10)){
79         cout << "Неверный ввод! " << endl;
80         return;
81     }
82
83     books[cnt++] = NewBook;
84     SaveFile("Books.txt", NewBook);
85
86
87
88
89
```

```

mymain.cpp x GUIDE.cpp Books.txt
practice_Book_Tracker > mymain.cpp > ShowStat()
61 void AddBook(){
89
90     cout << "Книга добавлена!" << endl;
91 }
92
93
94 void ShowBooks(){
95     if (cnt == 0){
96         cout << "Нет книг" << endl;
97         return;
98     }
99
100     cout << " --- Все книги --- \n" << endl;
101     cout << left << setw(50) << "Название" << " | "
102         << setw(30) << "Автор" << " | "
103         << setw(15) << "Год прочтения" << " | "
104         << "Оценка \n";
105     cout << string(100, '-') << endl;
106
107     for (int i = 0; i < cnt; i++){
108         cout << left << setw(50) << books[i].name << " | "
109             << setw(30) << books[i].author << " | "
110             << setw(15) << books[i].data << " | "
111             << books[i].score << endl;
112         cout << string(100, '-') << endl;
113     }
114 }
115
116 void ShowStat(){
117     if (cnt == 0){
118         cout << "Не найдено статистики" << endl;
119         return;
120     }
121     float sr_znach = 0;
122     float sum_sc = 0;
123     float max_sc = books[0].score;
124
125     for (int i = 0; i < cnt; i++){
126
127         sum_sc += books[i].score;
128
129         if (max_sc < books[i].score){
130             max_sc = books[i].score;
131         }
132     }
133
134     sr_znach = sum_sc / cnt;

```

py.

Ln 133, 0

```
mymain.cpp x GUIDE.cpp Books.txt
practice_Book_Tracker > mymain.cpp > ShowStat()
116 void ShowStat(){
125     for (int i = 0; i < cnt; i++){
133
134         sr_znach = sum_sc / cnt;
135
136         cout << " " << endl;
137         cout << " --- Статистика --- " << endl;
138         cout << "Всего книг прочитано - " << cnt << endl;
139         cout << "Средняя оценка - " << sr_znach << endl;
140         cout << "Самая высокая оценка - " << max_sc << endl;
141     }
142
143
144 void Menu(){
145     cout << " " << endl;
146     cout << "Добро пожаловать в трекер книг!" << endl;
147     cout << "1. Добавить книгу" << endl;
148     cout << "2. Показать все книги" << endl;
149     cout << "3. Показать статистику" << endl;
150     cout << "4. Выход" << endl;
151     cout << "Выберите пункт меню - " << endl;
152 }
153
154 int main(){
155
156     LoadFile("Books.txt");
157
158     int choice;
159
160     do{
161         Menu();
162         cin >> choice;
163
164         switch (choice)
165         {
166             case 1:
167                 AddBook();
168                 break;
169             case 2:
170                 ShowBooks();
171                 break;
172             case 3:
173                 ShowStat();
174                 break;
175             case 4:
176                 cout << "Выход из программы" << endl;
177                 break;
```

```
mymain.cpp x GUIDE.cpp Books.txt
practice_Book_Tracker > mymain.cpp > ShowStat()
116 void ShowStat(){
142
143
144 void Menu(){
145     cout << " " << endl;
146     cout << "Добро пожаловать в трекер книг!" << endl;
147     cout << "1. Добавить книгу" << endl;
148     cout << "2. Показать все книги" << endl;
149     cout << "3. Показать статистику" << endl;
150     cout << "4. Выход" << endl;
151     cout << "Выберите пункт меню - " << endl;
152 }
153
154 int main(){
155     LoadFile("Books.txt");
156
157     int choice;
158
159     do{
160         Menu();
161         cin >> choice;
162
163         switch (choice)
164         {
165             case 1:
166                 AddBook();
167                 break;
168             case 2:
169                 ShowBooks();
170                 break;
171             case 3:
172                 ShowStat();
173                 break;
174             case 4:
175                 cout << "Выход из программы" << endl;
176                 break;
177             default:
178                 cout << "Неверный выбор" << endl;
179         }
180     } while (choice != 4);
181     return 0;
182 }
183
184
185
186
```

Ln 133, C

#### 4. Ссылка на репозиторий

<https://github.com/esreveR1162/book-tracker.git>

#### 5. Вывод

За время работы я освоил работу со структурами данных, с файлами, научился управлять интерфейсом и разбирать ошибки.