ФИО: Ефимюк Юрий Дмитриевич

Группа: СКБ 251

1. **Задание**

Написать консольную программу которая должна иметь:

1. Добавлять информацию о книге:

o Название (до 100 символов),

o Автор (до 50 символов),

o Год прочтения,

o 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:

o Количество прочитанных книг;

o Средняя оценка;

o Самая высоко оценённая книга.

**2. Описание алгоритма**

Программа протестирована на GNU LINUX

* Структура knig содержит параметры обьекта клааса – Название книги, имя автора, дату прочтения, оценка
* Rep – Функция для замены элемента n строки на элемент q
  + Принимает символ n и символ q
  + Возвращает измененную строку
* Writefl – функция реализующая запись в файл
  + Принимает название книги, имя автора, дату прочтения, оценку
* Cl – функция реализующая очистку экрана для создания красоты
* Readf – функция реализующая чтение из файла
* Принимает файл откуда читать и аргумент проверку на чтение
* Compare – функция реализующая сравнения для обьекта pair
* Принимает пару чисел a и пару чисел b
* Возвращает первый элемент пары а>первого элемента пары б
* Getstat- функция реализующая получение статистики из файла
  + Ищет Книги с максимально возможным рейтингом
  + Выводит среднюю оценку
* Start – фунция стартового окна который грубо говоря меню
* Func – фунция отвечающая за реализацию в коде всех используемых функций (getstat, readf, cl, compare, rep)

**3. Код**

//ошибки плохо работает система обработки ошибки на ввод даты  
//уходит в бесконечный цикд  
  
  
#include "iostream"  
#include "string"  
#include "fstream"  
#include "vector"  
#include "sstream"  
#include "algorithm"  
#include "cstdlib"  
#include "limits"  
  
//#define DEBUG  
using namespace std;  
char\* sys;  
vector<pair<int, string>> nameknmark;//массив с парами типа (оценка книги; название книги)  
  
  
struct knig {//структура отвечающая за класс книг которая принимает на вход имя книги, автора, дату написания, оценку  
 string name, author;  
 int mark = 0;  
 int date = 0;  
  
};  
  
  
  
string rep(string h,char from,char to){  
 string ot="";  
 for (auto x: h){  
 if (x==from){  
 ot.push\_back(to);  
  
 }else{  
 ot.push\_back(x);  
 }  
 }  
 return ot;  
}  
  
//структура отвечающая за запись параметров структуры knig в файл  
  
void writefl(string n, string a, int d, int m) {  
 ofstream out("books.txt", ios::app);  
 if (out.is\_open()) {  
 out << n << " " << a << " " << d << " " << m << "\n";  
 }  
 out.close();  
}  
  
void cl(){  
 cin.ignore();  
 cin.get();  
 system(sys);  
}  
  
  
//чтение праарметров каждой из книги в файл  
  
int readf(string namef, int k) {  
 string line;  
 string n, a;  
 int m, d;  
 ifstream in(namef);  
  
 if (in.is\_open()) {  
 if (k==1){  
 cout<<"\033[31m"<< "Название ";  
 cout<<"\033[32m"<<"Автор ";  
 cout<<"\033[33m"<<"Дата прочтения ";  
 cout<<"\033[34m"<<"Дата прочтения "<<endl;  
 cout << "\033[0m";  
 cout<< "===================================="<<endl;  
 }  
  
 while (getline(in, line)) {  
 stringstream ss(line);  
 ss >> n >> a >> d >> m;  
 if (k == 1) {  
 cout<<"\033[31m"<< rep(n,'\_',' ') ;  
 cout<<"\033[32m"<<" "<<rep(a,'\_',' ');  
 cout<<"\033[33m"<<" " <<d;  
 cout<<"\033[34m"<< " "<< m <<endl;  
 cout << "\033[0m";  
 }  
 nameknmark.push\_back({m, rep(n,'\_',' ')});  
  
 }  
 }else{  
 cout<<"Добавьте книги!"<<endl<<endl;  
 cout<<"Нажмите Enter для продолжения"<<endl;  
 cl();  
 return 1;  
 }  
 if (k==1){  
 cout<<endl;  
 cout<<"Нажмите Enter для продолжения";  
 cl();  
 }  
 in.close();  
 return 0;  
}  
  
  
//функция сравнения для sort  
bool compare(pair<int, string> a, pair<int, string> b) {  
 return a.first > b.first;  
}  
  
  
//получение статистики из книг  
  
void getstat() {  
 int j=readf("books.txt",0);  
 if (j==1){  
 return;  
 }  
 cout << "Количество прочитанных книг: " << nameknmark.size() << endl;  
 sort(nameknmark.begin(), nameknmark.end(), compare);  
 float k = 0;  
 int ch = 0;  
 for (auto x: nameknmark) {  
 string n;  
 k += x.first;  
 ch += 1;  
 }  
 cout << "Средняя оценка: " << k / static\_cast<float>(ch) << endl;  
 int sch = 0;  
 cout << "Топ произведений: " << endl;  
 do {  
  
 cout << nameknmark[sch].second << " " << nameknmark[sch].first << endl;  
 sch++;  
 } while (nameknmark[sch].first == nameknmark[sch - 1].first && sch < ch);  
 cout<<endl;  
 cout<<"Нажмите Enter для продолжения";  
 cl();  
}  
  
  
//обработка ввод с клавиатуры для проги  
  
void func(int key) {  
 nameknmark.resize(0);  
 knig bobrik;  
 switch (key) {  
 case 1: {  
 do {  
 cout << "Название (до 100 символов)\n";  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 getline(cin,bobrik.name);  
 bobrik.name=rep(bobrik.name,' ','\_');  
 if ((bobrik.name).length() >= 100) {  
 system(sys);  
 cout << "Неверный формат" << endl;  
 cl();  
 continue;  
 }  
 break;  
 }while(1==1);  
 do{  
 metk2:  
 cout << "Автор (до 50 символов)\n";  
 getline(cin,bobrik.author);  
 bobrik.author=rep(bobrik.author,' ','\_');  
  
 if ((bobrik.author).length() >= 50) {  
 system(sys);  
 cout << "Неверный формат" << endl;  
 cl();  
 continue;  
 }  
 break;  
 }while(1==1);  
  
  
 do {  
  
  
 cout << "Год прочтения" << endl;  
 string s;  
 cin >> s;  
 try {  
 bobrik.date = stoi(s);  
 } catch (const std::invalid\_argument &e) {  
 system(sys);  
 cout << "Неверный формат" << endl;  
 cl();  
 continue;  
 }  
 if (bobrik.date >= 2026 || bobrik.date <= 0) {  
 system(sys);  
 cout << "Неверный формат" << endl;  
 cl();  
 continue;  
 }  
 break;  
 } while (1==1);  
 do {  
 cout << "Оценка по 10 бальной шкале\n";  
 string s1;  
 cin >> s1;  
 try {  
 bobrik.mark = stoi(s1);  
  
 } catch (const std::invalid\_argument &e) {  
 system(sys);  
 cout << "Неверный формат" << endl;  
 cl();  
 continue;  
 }  
  
 if (bobrik.mark > 10 || bobrik.mark < 0) {  
 system(sys);  
 cout << "Неверный формат" << endl;  
 cl();  
  
 continue;  
 }  
 break;  
 } while (1==1);  
#ifdef DEBUG  
 cout<<bobrik.name<<"\n"<<bobrik.author<<"\n"<<bobrik.date<<"\n"<<bobrik.mark;  
#endif  
 writefl(bobrik.name, bobrik.author, bobrik.date, bobrik.mark);  
 break;  
 }  
 case 2: {  
 readf("books.txt",1);  
 break;  
 }  
 case 3: {  
 getstat();  
 break;  
 }  
 }  
  
  
}  
  
void start() {  
 int a = 0;  
 system(sys);  
 while (a != 4) {  
 metk5:  
 cout << "=============########===============" << endl;  
 cout << "Для добавления книги нажмите 1\n";  
 cout << "Для показа списка книг нажмите 2\n";  
 cout << "Для просмотра статистики нажмите 3\n";  
 cout << "Для выхода нажмите 4\n";  
 string s2;  
 cin >> s2;  
 system(sys);  
 try {  
 a = stoi(s2);  
  
 } catch (const std::invalid\_argument &e) {  
 cout << "Неверный формат" << endl;  
 continue;  
 }  
 if (!(a > 0 && a < 5)) {  
 continue;  
 }  
 func(a);  
 cout << "\n";  
 }  
}  
  
int main() {  
 cout<<"Какая у вас система w-windows, l - linux"<<endl;  
 char buf[255];  
 string kl;  
 cin>>kl;  
 if(kl=="w"){  
 strcpy(buf,"CLS");  
 sys=buf;  
  
  
 }else{  
 strcpy(buf,"clear");  
 sys=buf;  
 }  
 cout<<endl;  
 start();  
}