МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА ИНФОРМАЦИОННО-СЕТЕВЫХ ТЕХНОЛОГИЙ

КУРСОВАЯ РАБОТА  
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. техн. наук |  |  |  | Курицын К.А. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ |
| КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ «БИБЛИОТЕКА» |
| по дисциплине: ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 1541 |  |  |  | Карандашев А.А. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2018

1. **Перечисление функциональных требований**

Консольное приложение «Библиотека» должно реализовывать управление электронным каталогом книг, хранящихся в библиотеке. Сам каталог является текстовым файлом, доступным для просмотра. Управление должно осуществляется через консольное приложение, включающее в себя функции добавления новой книги, поиск по жанру всех имеющихся книг, учёт количества книг в наличии на данный момент или же количество книг необходимых для заказа.

1. **Ограничения и иные функциональные требования**

Каждая книга должна содержать параметры: жанр, издательство, серия, год выпуска, количество страниц, переплёт, формат, автор, тираж.

Книги добавлять в библиотеку можно через консоль или через файл поштучно.

Результат записывается в файл формата txt, доступным для просмотра.

Записанные в каталог книги должны быть распределены по жанрам.

Код реализовать на языке С++, с использованием классов.

1. **Требования к оборудованию**

Консольное приложение.

1. **Требования к стороннему ПО**

Операционная система Windows 7 и Visual Studio 2017.

1. **Технические спецификации**

При запуске приложения перед пользователем отображается название приложения и варианты действий для работы с книгами.

Нулевой пункт отвечает за выход из приложения.

Первый пункт отвечает за добавление новой книги из консоли.

Второй пункт отвечает за добавление новой книги из файла.

Третий пункт отвечает за поиск всех названий книг по указанному жанру.

Четвёртый пункт отвечает за добавление новых экземпляров книг в каталог.

Пятый пункт отвечает за удаление экземпляров книг из каталога.

Если выбран первый вариант, то пользователь может ввести с консоли все данные о новой книге, после чего она автоматически добавляется в каталог в файле, о чём приходит сообщение на экран. Затем программа вновь спрашивает хотим ли мы добавить книгу и каким способом.

Если выбран второй вариант, то из файла подгрузки автоматически загружается в библиотеку необходимый образец книги, о чём пользователю сообщается на экране.

Затем программа вновь спрашивает хотим ли мы добавить книгу и каким способом.

При выборе загрузки из консоли, приложение очищает экран и последовательно запрашивает пользователя характеристики новой книги. В случае если такая книга уже была – выводиться сообщение что книга уже есть в каталоге. Далее ожидается нажатие Enter после чего пользователь вновь попадает в основное меню.

При выборе поиска, запрашивается интересующий нас жанр, после чего выводятся все книги данного жанра и их количество на складе. Ожидается нажатие Enter после чего пользователь вновь попадает в основное меню.

При выборе добавления или удаления экземпляров книги, консоль опрашивает пользователя серию книги, и количество в котором изменяем. После чего ожидается нажатие Enter после чего пользователь вновь попадает в основное меню.

1. **Взаимодействия с внешними системами**

Отсутствуют.

1. **Взаимодействия с внутренними системами**

Отсутствуют.

1. **Программа и методика испытаний**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Сценарий проверки | Ожидаемый результат | Результат |
| 1 | Включение приложения | Отображение названия программы и всех пунктов для работы с файлом каталога:  0)Выход  1)Добавление из консоли  2) Добавление из файла  3) Поиск по жанру  4)Добавление экземпляра  5)Удаление экземпляра | Готово |
| 2 | В меню выбрать 0 | Выход из приложения | Готово |
| 3 | В меню выбрать 1 | Добавление книги из консоли и запись в каталог | Готово |
|  |  | В случае повтора выводится предупреждение, что такая книга уже есть | Готово |
| 4 | В меню выбрать 2 | Добавление книги | Готово |
| 5 | В меню выбрать 3 | Запрос на поиск по жанру | Готово |
|  |  | Выдача всех актуальных результатов из базы в консоль | Готово |
| 6 | В меню выбрать 4 | Запрос на серию книги и количество добавляемых книг | Готово |
|  |  | Изменение файла каталога | Готово |
| 7 | В меню выбрать 5 | Запрос на серию книги и количество убранных книг | Готово |
|  |  | Изменение файла каталога | Готово |
| 8 | Возможность просмотра файла каталоги, рабочего файла | Отображение актуальной информации | Готово |
| 9 | Возможность изменение файла добавления книги и использование его в процессе работы с консолью | Добавление не приводящее к ошибкам | Готово |

1. **Листинг**

Ссылка на проект в Github: https://github.com/Alexander366/BibKyrsach

#include "stdafx.h"

#include <iostream>

#include <fstream>

#include <string>

#include <cstdlib>

#include <sstream>

#include <stdio.h>

#include <stdlib.h>

using namespace std;

class Kniga {

char Nazv[256];

char Ganr[256];

char Izdatel[256];

int Seriya;

int God;

int Str;

bool Pereplet;

char Format[256];

char Avtor[256];

int Tiraj;

int Kolichestvo;

public:

void INCONS(void)

{

int estlikn=0;

string ser;

cout << endl << "Название книги: ";

cin >> Kn.Nazv;

cout << endl << "Жанр книги: ";

cin >> Kn.Ganr;

cout << endl << "Издатель книги: ";

cin >> Kn.Izdatel;

cout << endl << "Серия выпуска книги: ";

cin >> Kn.Seriya;

cout << endl << "Год выпуска книги: ";

cin >> Kn.God;

cout << endl << "Количество страниц в книге: ";

cin >> Kn.Str;

cout << endl << "Переплёт книги: ";

cin >> Kn.Pereplet;

cout << endl << "Формат книги: ";

cin >> Kn.Format;

cout << endl << "Автор книги: ";

cin >> Kn.Avtor;

cout << endl << "Тираж: ";

cin >> Kn.Tiraj;

cout << endl << "Количество на складе: ";

cin >> Kn.Kolichestvo;

//ostringstream st;

//st.str();

//ser = st.str(Kn.Seriya);

estlikn=NALKN(Kn.Seriya);

if (estlikn == 0) {

fstream fin;

ofstream fout;

fin.open("Katalog.txt", ios::in);

fout.open("Work.txt", ios::out);

string line;

string buf = Kn.Ganr;

while (getline(fin, line)) // Пока мы получили строку - тело цикла исполняется || когда файл закончился (все строки считались) - цикл false

{

if ((line == "Fantasy") && (buf == "Fantasy"))

{

fout << "Fantasy" << "\n";

fout << Kn.Nazv << "\n";

//fout << Kn.Ganr << "\n";

fout << Kn.Izdatel << "\n";

fout << Kn.Seriya << "\n";

fout << Kn.God << "\n";

fout << Kn.Str << "\n";

fout << Kn.Pereplet << "\n";

fout << Kn.Format << "\n";

fout << Kn.Avtor << "\n";

fout << Kn.Tiraj << "\n";

fout << Kn.Kolichestvo << "\n";

fout << "==================================" << "\n";

}

if ((line == "Classic") && (buf == "Classic"))

{

fout << "Classic" << "\n";

fout << Kn.Nazv << "\n";

//fout << Kn.Ganr << "\n";

fout << Kn.Izdatel << "\n";

fout << Kn.Seriya << "\n";

fout << Kn.God << "\n";

fout << Kn.Str << "\n";

fout << Kn.Pereplet << "\n";

fout << Kn.Format << "\n";

fout << Kn.Avtor << "\n";

fout << Kn.Tiraj << "\n";

fout << Kn.Kolichestvo << "\n";

fout << "==================================" << "\n";

}

else

{

fout << line << "\n";

}

}

fin.close();

fout.close();

fin.open("Work.txt", ios::in);

fout.open("Katalog.txt", ios::out);

while (getline(fin, line))

{

fout << line << "\n";

}

fin.close();

fout.close();

}

else

{

cout << endl << "Такая книга уже есть в каталоге" << endl;

}

};

void INFILE(void)

{

fstream fin;

ofstream fout;

fstream fadd;

string buf;

fadd.open("Add.txt", ios::in);

for (int i=0; i<2; i++)

{

getline(fadd, buf);

}

fadd.close();

fin.open("Katalog.txt", ios::in);

fout.open("Work.txt", ios::out);

fadd.open("Add.txt", ios::in);

string line;

while (getline(fin, line)) // Пока мы получили строку - тело цикла исполняется || когда файл закончился (все строки считались) - цикл false

{

if ((line == "Fantasy") && (buf == "Fantasy"))

{

fout << "Fantasy" << "\n";

if (fadd)

{

string buf2;

int a = 0;

while (getline(fadd, buf2))

{

if(a!=1)fout << buf2 << "\n";

a++;

}

}

}

if ((line == "Classic") && (buf == "Classic"))

{

fout << "Classic" << "\n";

if (fadd)

{

string buf2;

int a = 0;

while (getline(fadd, buf2))

{

if (a != 1)fout << buf2 << "\n";

a++;

}

}

}

else

{

fout << line << "\n";

}

}

fin.close();

fout.close();

fadd.close();

fin.open("Work.txt", ios::in);

fout.open("Katalog.txt", ios::out);

while (getline(fin, line))

{

fout << line << "\n";

}

fin.close();

fout.close();

};

void FOUND(void)

{

string ganr;

string line;

int i = 0;

bool Flag = 0;

cout << endl << "Название жанра: ";

cin >> ganr;

fstream fin;

fin.open("Katalog.txt", ios::in);

while (getline(fin, line))

{

if (line == ganr) Flag = 1;

if (line == "----------------------------------") Flag = 0;

if(Flag==1)

{

if (i == 1) cout << "Название книги:" << line <<" ";

if (i == 10) cout << "В наличии: " << line <<"\n";

i++;

if (i == 11) i = 0;

}

}

fin.close();

};

int NALKN(int seriya)

{

fstream fin;

fin.open("Katalog.txt", ios::in);

string line;

char sr[256];

int i = 0;

\_itoa\_s(seriya, sr, 10);

while (getline(fin, line))

{

if (sr == line)return 1;

}

return 0;

fin.close();

};

void IZMENEN(int deist)

{

string sernum;

int newkol;

cout << endl << "Введите серийный номер выпуска книги: ";

cin >> sernum;

if (deist == 0)

{

cout << endl << "Введите количество добавляемых книг: ";

cin >> newkol;

}

else

{

cout << endl << "Введите количество взятых книг: ";

cin >> newkol;

}

fstream fin;

ofstream fout;

fin.open("Katalog.txt", ios::in);

fout.open("Work.txt", ios::out);

string line;

bool Flag=0;

int i = 0;

int kolich;

while (getline(fin, line)) // Пока мы получили строку - тело цикла исполняется || когда файл закончился (все строки считались) - цикл false

{

if (line == sernum) Flag = 1;

if (Flag == 1) i++;

if (i == 8)

{

istringstream buf(line);

buf >> kolich;

if (deist == 0) kolich = kolich + newkol;

if (deist == 1) kolich = kolich - newkol;

line = to\_string(kolich);

i = 0;

Flag = 0;

}

fout << line << "\n";

}

fin.close();

fout.close();

fin.open("Work.txt", ios::in);

fout.open("Katalog.txt", ios::out);

while (getline(fin, line))

{

fout << line << "\n";

}

fin.close();

fout.close();

};

}Kn;

int main()

{

setlocale(LC\_ALL, "Russian");

int deistvie;

while (1)

{

cout << "ПРИЛОЖЕНИЕ БИБЛИОТЕКА" << endl << endl;

cout << "Выбирите операцию:" << endl <<

"0) Закрыть" << endl <<

"1) Добавление книги в библиотеку из консоли" << endl <<

"2) Добавление книги в библиотеку из файла" << endl <<

"3) Поиск по жанру" << endl <<

"4) Добавление книги на склад" << endl <<

"5) Удаление книги со склада" << endl <<endl;

cin >> deistvie;

if (deistvie == 0) return 0;

if (deistvie == 1) { system("cls"); Kn.INCONS(); getchar(); getchar();system("cls");}

if (deistvie == 2) { system("cls"); Kn.INFILE(); getchar(); getchar();system("cls");}

if (deistvie == 3) { system("cls"); Kn.FOUND(); getchar(); getchar();system("cls");}

if (deistvie == 4) { system("cls"); Kn.IZMENEN(0); getchar(); getchar(); system("cls"); }

if (deistvie == 5) { system("cls"); Kn.IZMENEN(1); getchar(); getchar(); system("cls"); }

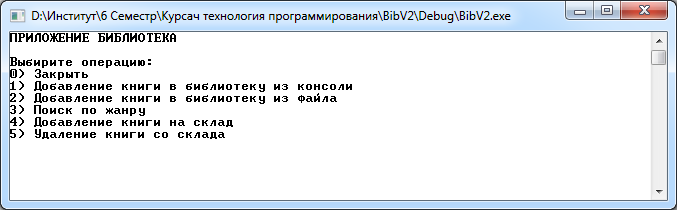
}

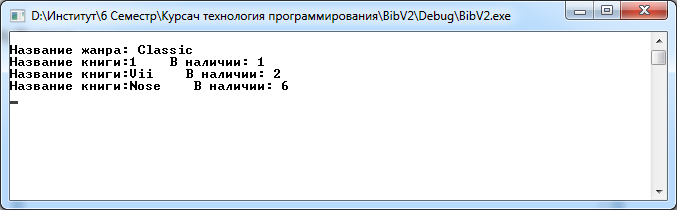
system("pause");

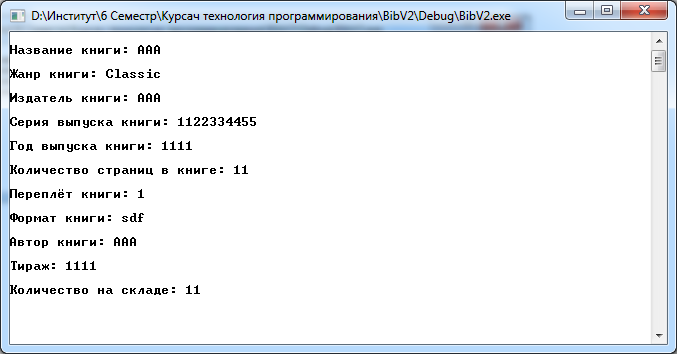
return 0;

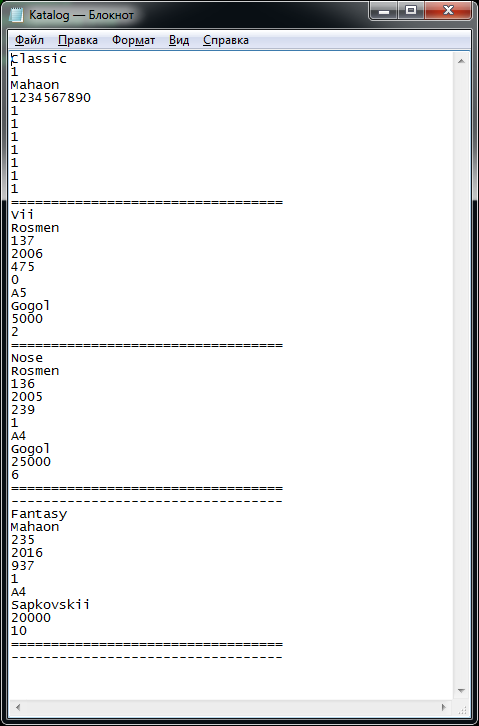
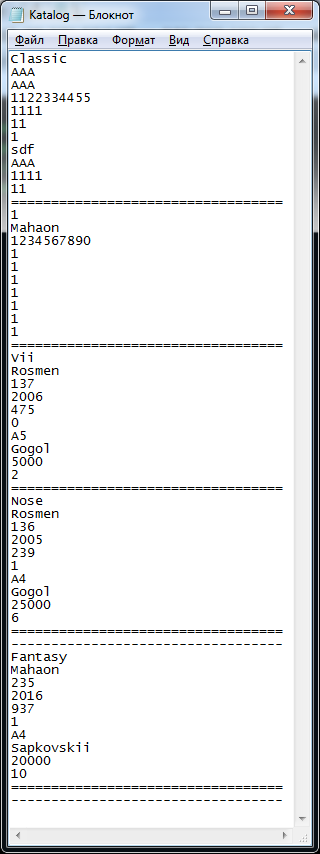
}

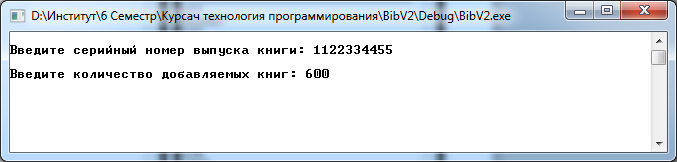
1. **Вывод**

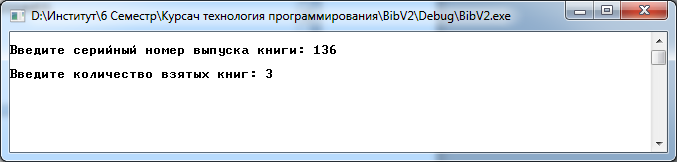


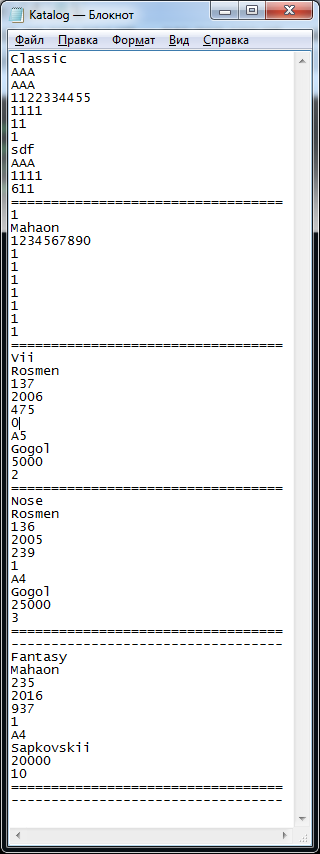
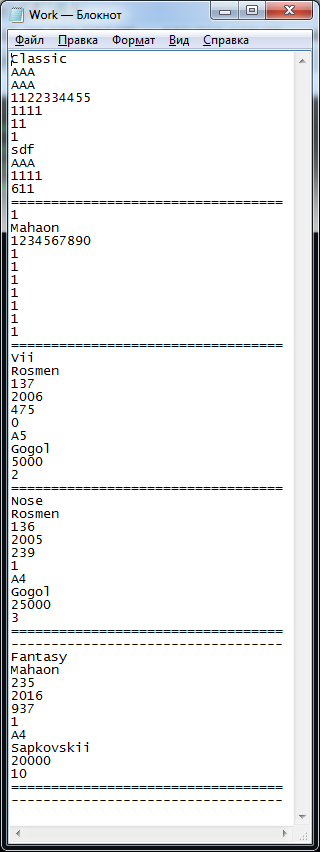


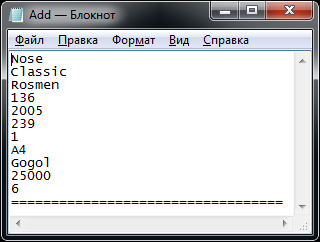








1. **Литература**

* https://support.microsoft.com
* http://www.cyberforum.ru
* https://ru.stackoverflow.com
* http://cppstudio.com
* http://www.c-cpp.ru
* Б. Керниган и Д. Ритчи – Язык программирования Си
* Шилдт Г. Самоучитель С++
* Бьерн Страуструп - Язык программирования С++