

КОМПЬЮТЕРНАЯ АКАДЕМИЯ ШАГ
МАЛАЯ КОМПЬЮТЕРНАЯ
АКАДЕМИЯ



ИТОГОВЫЙ ПРОЕКТ НА ТЕМУ «БИБЛИОТЕКА»

По курсу «Процедурное программирование на С»



Исполнитель:
Студент группы ИТ-16-5
Твердохлеб Владислав
Юрьевич

ХАРЬКОВ
2017 г.

ОГЛАВЛЕНИЕ

<u>ПОСТАНОВКА ЗАДАЧИ -></u>	2
<u>ПРОЕКТИРОВАНИЕ СИСТЕМЫ -></u>	3
<u>ЗАДАЧИ ПРОЕКТА -></u>	4
<u>СРАВНЕНИЕ -></u>	5
<u>НЕДОСТАТКИ -></u>	6
<u>ПОЛЬЗОВАТЕЛИ ПРОЕКТА -></u>	6
<u>АРХИТЕКТУРА ПРОЕКТА -></u>	7
<u>ФУНКЦИОНАЛ ПРОЕКТА -></u>	10

ПОСТАНОВКА ЗАДАЧИ ->

Была поставлена задача создать программную систему, предоставляющая возможность посетителям посмотреть любую информацию о изданиях в библиотеке. Издания распределяются на книги, журналы, газеты, энциклопедии и словари. Посетители имеют возможность детально изучить каждое издание, посмотрев его полное название, автора, издательство, жанр, тематику, количество страниц и год публикации.

Также в системе должна использоваться функция для просмотра информации о новых полученных изданиях за каждый месяц.

Так как библиотека получает издания каждый день, то нужно сделать обязательную админ-панель для заведующего-библиотекаря. Функции, такие как, добавить и удалить издания в базу являются обязательными для реализации.

Очень важная деталь в функционале также – функции сортировки. Следует сделать сортировки изданий по критериям |А-Я| и |MIN-MAX|.

Программа должна быть оформлена в эргономичном интерфейсе, а также иметь сопроводительную документацию. Должны присутствовать вынесение функций и процедур в отдельные файлы (header files) и сохранения результатов с последующим их использованием (текстовый или бинарный файл).

Использование динамического массива структур также не должно оставаться в стороне.

ПРОЕКТИРОВАНИЕ СИСТЕМЫ ->





ПОЛЬЗОВАТЕ
ЛИ

Логин (char[])
Пароль (char[])

ИЗБРАННОЕ

Название (char[])
Тип (int)

ЗАДАЧИ ПРОЕКТА ->

Данная система была создана для поиска информации о любом издании в библиотеке. Благодаря высокой эргономики, посетители могут максимально быстро находить интересующую им информацию.

Одна из идей автора такова, что система может использоваться в терминалах или же в компактных ноутбуках, стоящих в библиотеке. Вместо того, чтобы стоять очередями с вопросами к библиотекарю - он становится одним из администраторов программы, а каждый посетитель сможет без нервозности найти ему интересующую книгу/журнал/газету/энциклопедию/словарь. Всего в два клика можно посмотреть свежесть и дату завоза журнала или газеты, или проверить наличие очередной книги любимого автора.



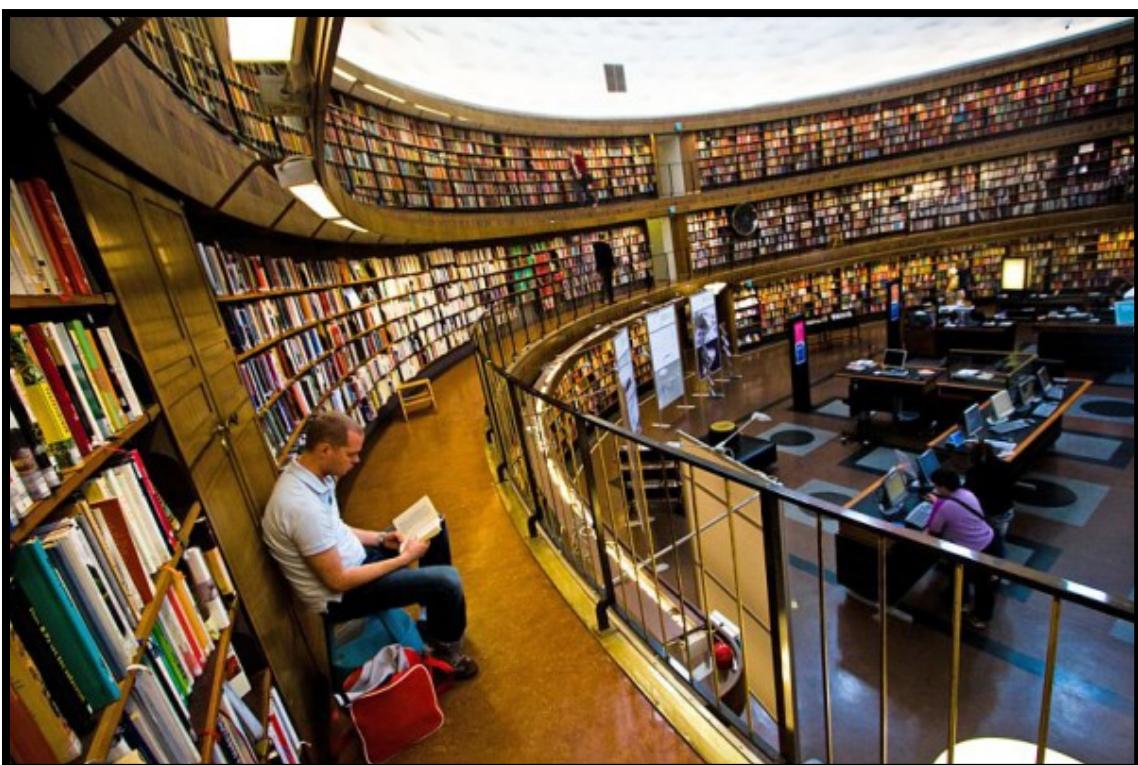
Роль библиотекаря тоже останется, он будет проверять задолженности, выдавать издания посетителям, следить за всей базой данных библиотеки и по надобности добавлять или

удалять издания. Но основную работу будет совершать данная система.

БИБЛИОТЕКА, НЕ ИМЕЮЩАЯ ТАКУЮ ПРОГРАММНУЮ СИСТЕМУ У СЕБЯ В РАСПОРЯЖЕНИИ ->



БИБЛИОТЕКА, ОБОРУДОВАННАЯ ТАКОЙ СИСТЕМОЙ ->



НЕДОСТАТКИ ->

- При аварийном отключении электричества могут произойти различные нехорошие последствия (но это бывает редко, на такой случай хорошо оборудованные библиотеки имеют при себе ИБП):
 - Ошибки программы в дальнейшем её использовании
 - Потеря n-ого кол-ва зарегистрированных аккаунтов
 - Потеря n-ой части базы данных
- Множеству пожилым людям, пенсионерам, после такого обновления библиотеки станет проблематично делать ранние вещи. Им придется осваивать новые технологии.

НЕДОСТАТКИ ПРОГРАММЫ ->

- Каждое издание не имеет своего персонального кода
- Каждое издание не имеет описания к нему

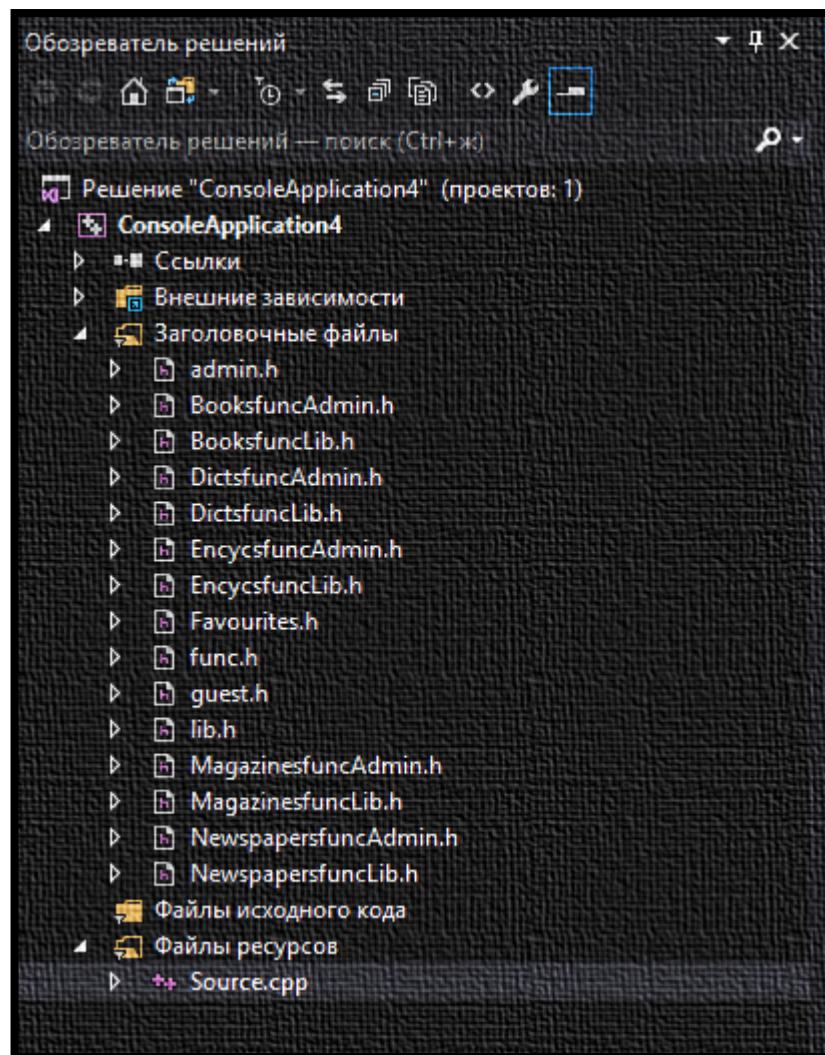
ПОЛЬЗОВАТЕЛИ ПРОЕКТА ->

- ПРОФИЛЬ «ГОСТЬ»
- ПРОФИЛЬ «ЗАРЕГИСТРИРОВАННЫЙ»
- ПРОФИЛЬ «АДМИНИСТРАТОР»

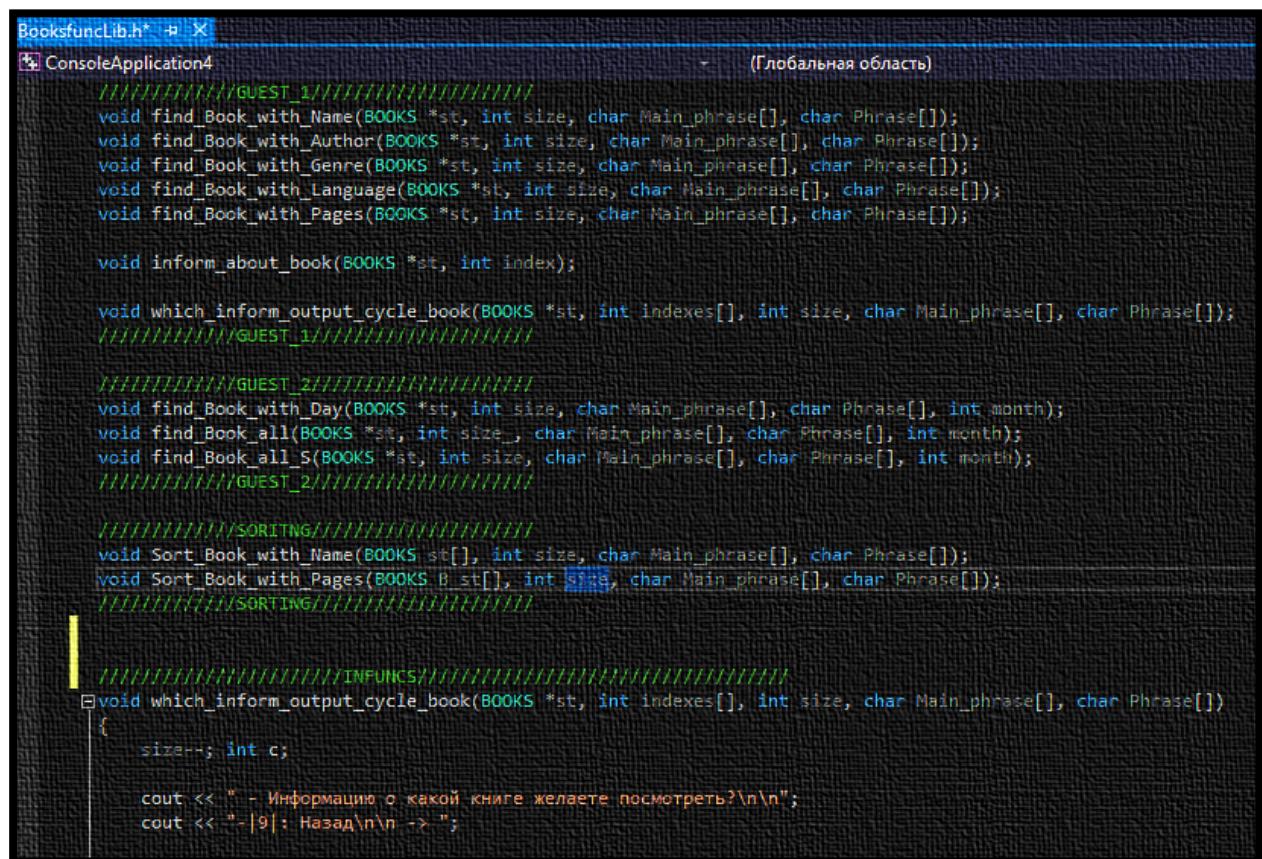
АРХИТЕКТУРА ПРОЕКТА ->

Весь код отсортирован по отдельным файлам (headers files), присутствует множество пометок

Обозреватель решений:



Пример одного заголовочного файла:



BooksfuncLib.h* → X

ConsoleApplication4 (Глобальная область)

```
|||||||//GUEST_1|||||||||||||||
void find_Book_with_Name(BOOKS *st, int size, char Main_phrase[], char Phrase[]);
void find_Book_with_Author(BOOKS *st, int size, char Main_phrase[], char Phrase[]);
void find_Book_with_Genre(BOOKS *st, int size, char Main_phrase[], char Phrase[]);
void find_Book_with_Language(BOOKS *st, int size, char Main_phrase[], char Phrase[]);
void find_Book_with_Pages(BOOKS *st, int size, char Main_phrase[], char Phrase[]);

void inform_about_book(BOOKS *st, int index);

void which_inform_output_cycle_book(BOOKS *st, int indexes[], int size, char Main_phrase[], char Phrase[]);
|||||||//GUEST_1|||||||||||||||
|||||||//GUEST_2|||||||||||||||
void find_Book_with_Day(BOOKS *st, int size, char Main_phrase[], char Phrase[], int month);
void find_Book_all(BOOKS *st, int size, char Main_phrase[], char Phrase[], int month);
void find_Book_all_S(BOOKS *st, int size, char Main_phrase[], char Phrase[], int month);
|||||||//GUEST_2|||||||||||||||
|||||||//SORTING|||||||||||||||
void Sort_Book_with_Name(BOOKS st[], int size, char Main_phrase[], char Phrase[]);
void Sort_Book_with_Pages(BOOKS B st[], int size, char Main_phrase[], char Phrase[]);
|||||||//SORTING|||||||||||||||
|||||||//INFUNCS|||||||||||||||
void which_inform_output_cycle_book(BOOKS *st, int indexes[], int size, char Main_phrase[], char Phrase[])
{
    size-- ; int c;

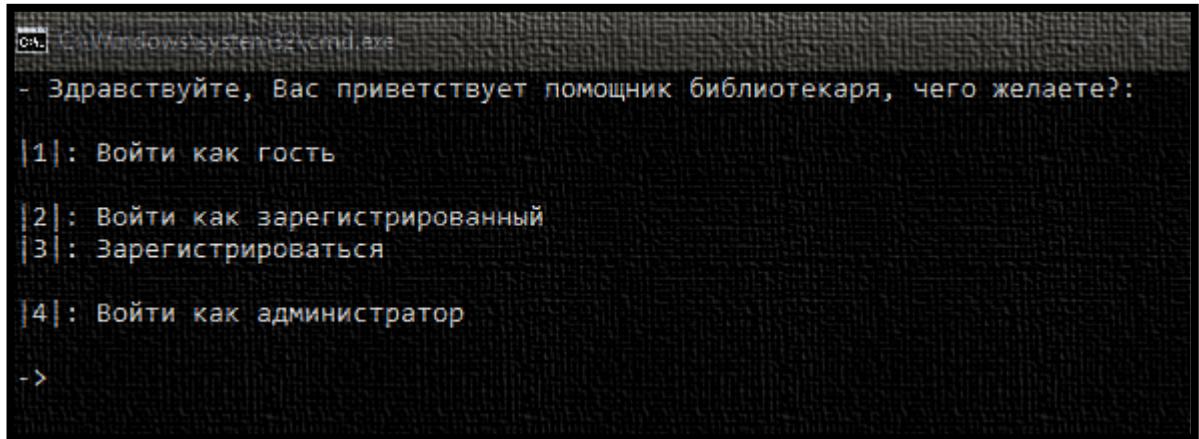
    cout << " - Информацию о какой книге желаете посмотреть?\n\n";
    cout << "-[9]: Назад\n\n -> ";
}
```

Меню выполнено оператором меток goto. Примеры МЕТОК:

```
guest:  
  
    system("cls");  
    SetColor(8, 0);  
    cout << Main_phrase << endl;  
    SetColor(7, 0);  
  
    cout << " [1]: Информация о всех изданиях\n";  
    cout << " [2]: Информация о завозе за последний месяц\n";  
  
    if (mode == 2)  
    {  
        cout << " [3]: Меню сортировки\n";  
        cout << " [4]: Избранное\n";  
    }  
  
    cout << "\n-[0]: EXIT\n\n -> ";  
  
    cin >> choice; cin.get();  
  
    if (choice == 1)  
        goto guest_1;  
    else if (choice == 2)  
        goto guest_2;  
  
    else if (choice == 0)  
        goto exit;  
  
    if (mode == 2)  
    {  
        if (choice == 3)  
            goto sort;  
        if (choice == 4)  
            goto guest_1;  
    }  
  
guest_1:  
    system("cls");  
    SetColor(8, 0);  
    cout << Main_phrase << endl;  
  
    cout << " [Информация о всех изданиях] -> \n\n";  
    SetColor(7, 0);  
  
    cout << " [1]: Книги\n";  
    cout << " [2]: Журналы\n";  
    cout << " [3]: Газеты\n";  
    cout << " [4]: Энциклопедии\n";  
    cout << " [5]: Словари\n\n";  
  
    cout << " [6]: Вывести информацию о случайном издании\n\n";  
  
    cout << "-[9]: Назад\n";  
    cout << "-[0]: EXIT\n\n -> ";  
  
    cin >> choice; cin.get();  
  
    if (choice == 1)  
        goto books;  
    else if (choice == 2)  
        goto magazines;  
    else if (choice == 3)  
        goto newspapers;  
    else if (choice == 4)  
        goto encycs;  
    else if (choice == 5)  
        goto dicts;  
  
    else if (choice == 6)  
    {  
        inform_randomly(Books_st, Magazines_st, Newspapers_st, Encycs_st, Dicts_st, size_B, size_M, size_N, size_E, size_D);  
        goto guest_1;  
    }
```

ФУНКЦИОНАЛ ПРОЕКТА ->

Вид меню программы:

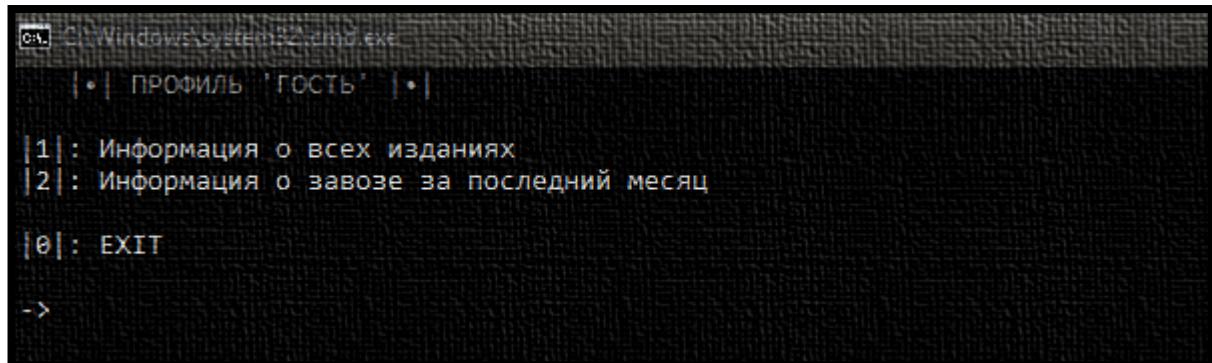


```
C:\Windows\system32\cmd.exe
- Здравствуйте, Вас приветствует помощник библиотекаря, чего желаете?:  
| 1 | : Войти как гость  
| 2 | : Войти как зарегистрированный  
| 3 | : Зарегистрироваться  
| 4 | : Войти как администратор  
->
```

Только что подключившийся к системе посетитель имеет возможность войти как гость, если он первый раз посещает библиотеку или редко в ней бывает. Для постоянных клиентов была сделана функция «Войти как зарегистрированный». Перед тем как это сделать они конечно регистрируются, регистрация проста - требуется ввести имя пользователя и пароль. Было бы глупо регистрироваться просто для просмотра информации, поэтому были реализованы две возможности для зарегистрированных. Первая – это меню сортировки, вторая более интересная – меню избранное.

Также присутствует вход как администратор. Вход для администратора был усложнен для безопасности – после 3 неправильно введенных данных графа входа в админ-панель блокируется. Администратору была добавлена возможность удалить пользователя и изменить значение последнего месяца.

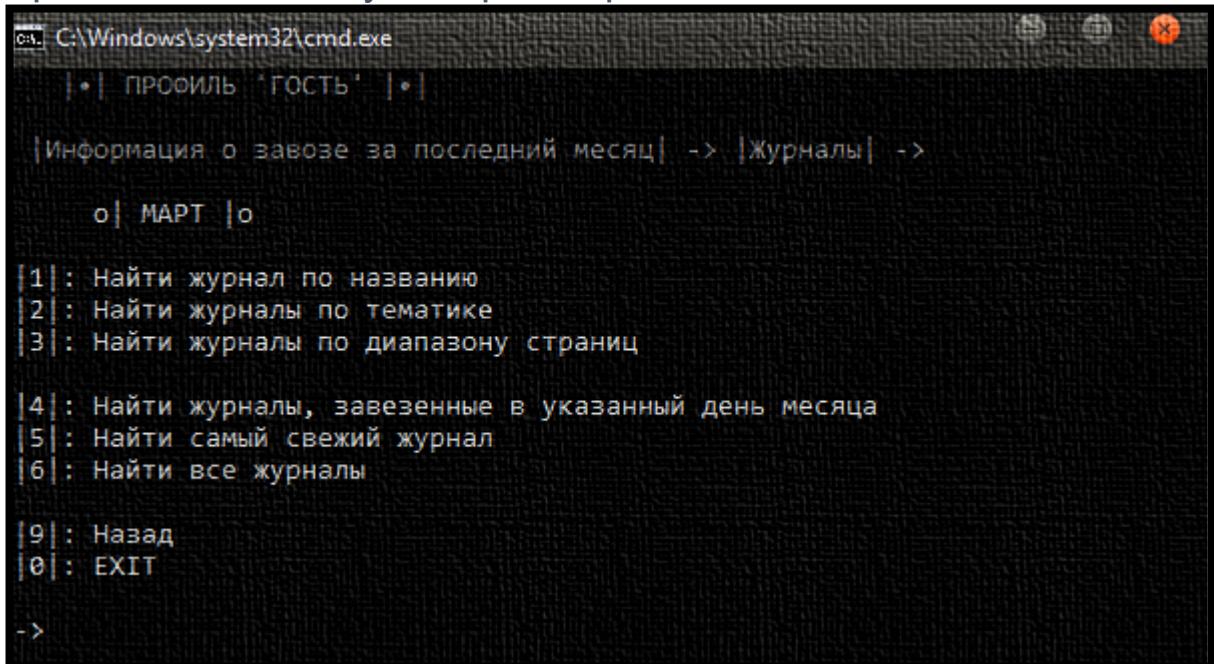
Изначально в профиле «Гость» есть две возможности:



```
C:\Windows\system32\cmd.exe
| • | ПРОФИЛЬ 'Гость' | • |
| 1 | : Информация о всех изданиях  
| 2 | : Информация о завозе за последний месяц  
| 0 | : EXIT  
->
```

посмотреть информацию о всех изданиях или о завозе за последний месяц, они и есть главные в программе. После входа в профиль кнопка выхода будет доступна всегда.

Кроме просмотра информации о изданиях, посетители также могут воспользоваться мелкими функциями, такими как: вывести информацию о случайном издании (Информация о всех изданиях), найти издания, завезенные в указанный день месяца, вывести самый свежий журнал или газету. Поиск производится без учета регистра.



C:\Windows\system32\cmd.exe

| • | ПРОФИЛЬ 'ГОСТЬ' | • |

| Информация о завозе за последний месяц | -> | Журналы | ->

о | МАРТ | о

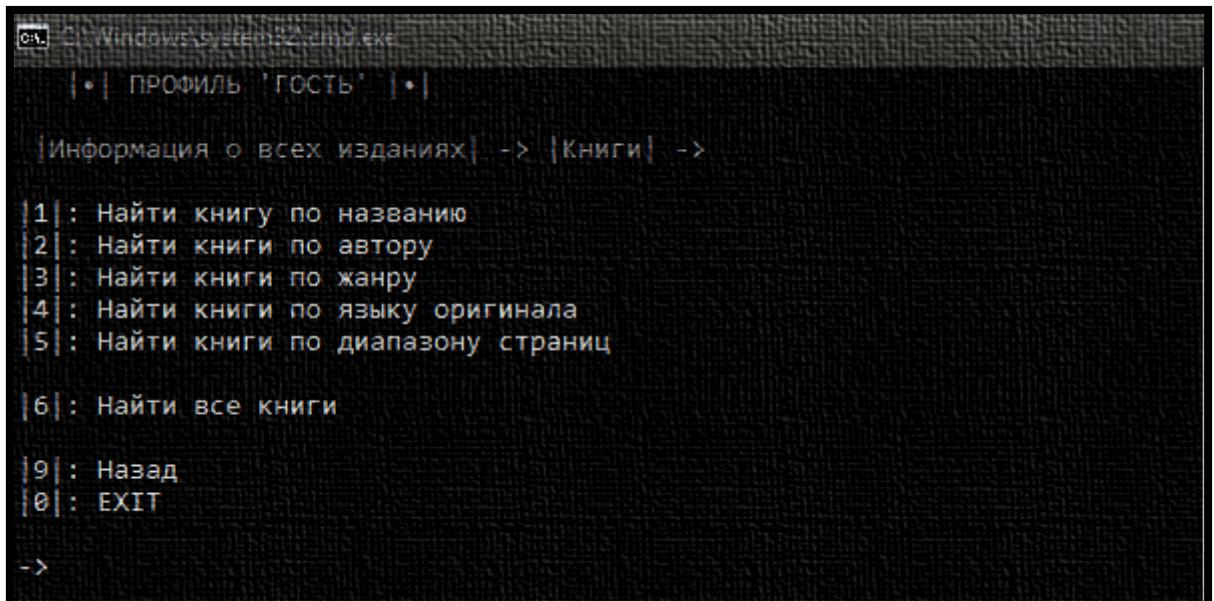
|1| : Найти журнал по названию
|2| : Найти журналы по тематике
|3| : Найти журналы по диапазону страниц

|4| : Найти журналы, завезенные в указанный день месяца
|5| : Найти самый свежий журнал
|6| : Найти все журналы

|9| : Назад
|0| : EXIT

->

Критерии поиска книги:



C:\Windows\system32\cmd.exe

| • | ПРОФИЛЬ 'ГОСТЬ' | • |

| Информация о всех изданиях | -> | Книги | ->

|1| : Найти книгу по названию
|2| : Найти книги по автору
|3| : Найти книги по жанру
|4| : Найти книги по языку оригинала
|5| : Найти книги по диапазону страниц

|6| : Найти все книги

|9| : Назад
|0| : EXIT

->

В проекте реализованы 3 вида сортировки: выбором, вставками и пузырьковая

Пузырьковая сортировка слов (А-Я):

```
for (int i = 1; i < size; i++) // пузырьком
{
    for (int j = 0; j < size - i; j++)
    {
        if (strcmp(arr[j + 1], arr[j]) < 0)
        {
            temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
        }
    }
}
```

Сортировка чисел выбором (MIN-MAX):

```
int tmp; int pos;

for (int i = 0; i < size; ++i) // выбором
{
    pos = i;
    tmp = st[i].pages;
    for (int j = i + 1; j < size; ++j)
    {
        if (st[j].pages < tmp)
        {
            pos = j;
            tmp = st[j].pages;
        }
    }
    st[pos].pages = st[i].pages;
    st[i].pages = tmp;
}
```

Обработка и сохранение данных происходит через бинарные файлы, в основном. В month.txt хранится значение последнего месяца.

У каждого зарегистрированного пользователя имеется бинарный файл типа «логин».dat. В случае удаления пользователя администратором такой файл удаляется.

Также каждый тип издания имеет собственный бинарный файл для хранения данных.

Рабочая папка проекта:

	Debug	24.04.2017 22:04	Папка с файлами
	admin.h	24.04.2017 1:59	C/C++ Header
	Admins.dat	21.04.2017 14:53	Файл "DAT"
	Books.dat	19.04.2017 22:12	Файл "DAT"
	BooksfuncAdmin.h	22.04.2017 20:35	C/C++ Header
	BooksfuncLib.h	24.04.2017 20:16	C/C++ Header
	ConsoleApplication4.vcxproj	21.04.2017 19:14	VC++ Project
	ConsoleApplication4.vcxproj.filters	21.04.2017 19:14	VC++ Project Filter...
	Dicts.dat	19.04.2017 21:29	Файл "DAT"
	DictsfuncAdmin.h	21.04.2017 17:06	C/C++ Header
	DictsfuncLib.h	22.04.2017 22:11	C/C++ Header
	Encycs.dat	19.04.2017 21:28	Файл "DAT"
	EncycsfuncAdmin.h	21.04.2017 17:04	C/C++ Header
	EncycsfuncLib.h	24.04.2017 22:33	C/C++ Header
	Favourites.h	22.04.2017 22:16	C/C++ Header
	func.h	24.04.2017 1:53	C/C++ Header
	guest.h	24.04.2017 1:59	C/C++ Header
	lib.h	23.04.2017 1:02	C/C++ Header
	Magazines.dat	19.04.2017 10:19	Файл "DAT"
	MagazinesfuncAdmin.h	21.04.2017 16:54	C/C++ Header
	MagazinesfuncLib.h	22.04.2017 22:11	C/C++ Header
	month.txt	22.04.2017 21:29	Текстовый докум...
	Newspapers.dat	19.04.2017 21:27	Файл "DAT"
	NewspapersfuncAdmin.h	21.04.2017 17:09	C/C++ Header
	NewspapersfuncLib.h	24.04.2017 22:34	C/C++ Header
	Source.cpp	24.04.2017 1:53	C++ Source
	test.dat	22.04.2017 22:14	Файл "DAT"
	Users.dat	22.04.2017 22:14	Файл "DAT"

ОФОРМЛЕНИЕ ПРОЕКТА ->

В проект была добавлена функция SetColor для изменения цвета текста в консоли.

```
void SetColor(int text, int background)
{
    HANDLE StdOut = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(StdOut, (WORD)((background << 4) | text));
}
```

The screenshot shows a Windows command prompt window titled 'cmd.exe' with the path 'C:\Windows\system32'. Inside the window, the following interaction is visible:

```
C:\Windows\system32\cmd.exe
| • | ПРОФИЛЬ 'test' | • |
[Избранное] -> [Добавить издание] ->
Название издания: война и мир
- Книга: Война и мир
Добавить в избранное?
|1|: Да
|2|: Нет
->
```

С помощью этой функции была окрашена верхняя информация: записывающийся путь пользователя по меню и профиль, в котором он находится.

| • | ПРОФИЛЬ 'test' | • |

| • | ПРОФИЛЬ 'ГОСТЬ' | • |

И чтобы немного придать реалистичности, в программе в редких случаях использовалась функция Sleep.