

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 14

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Доц., к.т.н.		К. А. Курицын
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЁТ О КУРСОВОЙ РАБОТЕ
«ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ»

по дисциплине: «Технологии программирования»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	1742		Д.Д.Иванов
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2020

Оглавление:

1. Постановка задачи	3
1.1. Введение в цели и задачи программы	3
1.2. Спецификация	3
1.3. Принцип работы с файлами	4
1.4. Информация по использованному паттерну	4
1.5. Описание классов	4
2. Основная часть.....	5
2.1. Программа и методика испытаний	6
3. Листинг	6
4. Список использованной литературы	18
5. Примеры работы программы.....	18

1. Постановка задачи

Создать систему запросов «Сайт → Магазины (3 штуки) → Склад». Товар, который интересует клиента – путеводители по разным городам и странам, которые имеют ряд характеристик: название, издательство, год, число страниц, наличие фотографий, наличие карты, переплет, формат, автор, количество. Пользователь использует сайт для нахождения интересующего товара, затем запрос отправляется в магазины, затем на склад. Использовать паттерн «Цепочка обязанностей». Данные о товаре для сайта, магазинов и склада берутся из внешних входных файлов «site», «shop1», «shop2», «shop3» и «warehouse» соответственно.

1.1 Введение в цели и задачи программы.

- 1) Разработать систему запросов по принципу “Сайт→ Магазины (3 штуки) →Склад”.
- 2) Создать понятное пользовательское меню.
- 3) Полная информация о товарах должна браться из соответствующих файлов магазина.
- 4) Реализовать поиск товара по цепочке системы запросов и в случае его обнаружения искомого пользователем товара уведомить о его наличии.
- 5) В случае, если в результате поиска оказалось, что товара нет в наличии, необходимо осуществить заказ товара со склада или уведомить пользователя о невозможности совершения заказа.

1.2 Спецификация

- 1) Реализация запросов с помощью паттерна “Цепочка обязанностей”.
- 2) Работа программы начинается с главного меню, имитирующее работу пользователя с сайтом. В нём отображается полный каталог товаров, тут же можно совершить заказ.
- 3) Вывод всей информации, как и ввод, осуществляется через консоль.
- 4) - Информация для вывода в каталог загружаются из файла «site.txt».
- Информация о наличии товаров в одном из 3-х магазинов загружается из файлов «shop1.txt», «shop2.txt», «shop3.txt» соответственно.
- Информация о наличии товаров на складе загружается из файла «warehouse.txt».
- 5) Класс guide содержит поля, содержащие полный список характеристик каждого товара (путеводителя), а именно: название, издательство, год, число страниц, наличие фотографий, наличие карты, переплет, формат, автор, количество.
- 6) Как только пользователь выберет интересующий его товар, будет осуществлён его поиск в файлах, содержащих информацию о наличии товаров в магазинах. Если интересующий товар будет найден в одном из магазинов, в нем будет создан заказ. Если поиск не увенчается успехом, пользователь будет уведомлён об отсутствии интересующего его товара.

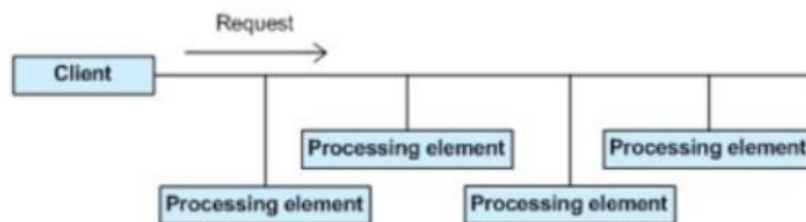
- 7) Если товар будет обнаружен в наличии на складе, будет произведен запрос о необходимости его бронирования на складе. Если поиск не увенчается успехом, пользователь будет уведомлён об отсутствии интересующего его товара.

1.3 Принцип работы с файлами:

- В данной работе используются файлы с расширением ".txt", указывающим на то, что это текстовый документ, который можно открыть в блокноте.
- Эти файлы являются базой данных для хранения информации о наличии и каталоге товаров.
- Файл site.txt должен содержать первую строку с перечислением всех параметров, последующие строки несут информацию по каждому из товаров. Параметры вводятся через пробел, переход к следующему товару через знак переноса строки.
- Файлы shop1.txt, shop2.txt, shop3.txt, warehouse.txt должны содержать первую строку с перечислением параметров, последующие строки со значениями двух параметров: артикул, количество. Параметры вводятся через пробел, переход к следующему товару через знак переноса строки.

1.4 Информация по использованному паттерну:

Основная суть паттерна:



Цепочка обязанностей — это поведенческий паттерн проектирования, который позволяет передавать запросы последовательно по цепочке обработчиков. Каждый последующий обработчик решает, может ли он обработать запрос сам и стоит ли передавать запрос дальше по цепи.

Паттерн “цепочка обязанностей” связывает в цепочку объекты-получатели, а затем передает запрос-сообщение от одного объекта к другому до тех пор, пока не достигнет объекта, способного его обработать.

Принцип работы данного паттерна в программе заключается в поиске искомого товара. Пользователь выбирает товар. По этому товару передается запрос по объектам (3 магазина + склад). Запрос передается по очереди, пока один из объектов не сможет его обработать (пока товар не будет найден в наличии в одном из магазинов/складе).

1.5 Описание классов:

В программе реализованы следующие классы: Catalog, Guide, Institution, Shop, Warehouse.

Класс Catalog содержит свойства и методы, необходимые для считывания информации из файла-каталога site.txt, также выводящие информацию пользователю в формате “сайта”.

Класс Guide содержит свойства и методы, необходимые для работы с ключевыми полями, такими как установка и возврат значения полей из каталога.

Класс Institution содержит важные методы по поиску интересующих покупателя товаров среди магазинов и склада.

Классы Shop и Warehouse являются классами наследниками класса INSTITUTION. В них описываются свойства и методы, необходимые для считывания информации о наличии товаров в магазинах и складе из соответствующих файлов.

2. Основная часть

2.1. Программа и методика испытаний

№	Сценарий проверки	Ожидаемый результат	Результат	№ спецификации
1	Запуск программы	Вывод на экран главного меню, имитирующего работу с сайтом.	Получен Рисунок 1	2
2	Необходимо осуществить заказ одного и того же товара несколько раз	Программа осуществляет поиск до первого найденного экземпляра. В первую очередь товар будет проверяться в наличии в магазинах, затем на складе.	Получен Рисунок 2	1
3	Запуск с файлами, лежащими в каталоге с проектом	Для того, чтобы сформировать каталог товаров программа использует файл «site.txt».	Получен Рисунок 3	3
4	Запуск	На экран выводится каталог товаров и полная информация о каждом из них.	Получен Рисунок 1	4

5	Осуществление заказа	Вывод на экран сообщения о успешной регистрации заказа в случае обнаружения интересующего путеводителя, в противном случае вывод на экран сообщения об отсутствии товара в наличии.	Получен Рисунок 2	5
6	Осуществление бронирования со склада	Вывод на экран запроса о совершении бронирования со склада, в случае положительного ответа производится бронь на складе, в противном случае вывод на экран сообщения об отмене заказа.	Получен Рисунок 4	6

2. Листинг

Ссылка на GitHub – <https://github.com/IvanovDaniil1998/kursovaya>

main.cpp

```
#include "catalog.h"
#include "shop.h"
#include "warehouse.h"

#include <iostream>
#include <string>
#include <windows.h>

using namespace std;

int main()
{
    system("chcp 1251>nul");
    string choise, temp;
    int ch;
    CATALOG *guide_catalog;
    SHOP *shop1;
    SHOP *shop2;
    SHOP *shop3;
    WAREHOUSE *warehouse1;

    guide_catalog = new CATALOG;
    shop1 = new SHOP("shop1.txt", "Обзор городов");
    shop2 = new SHOP("shop2.txt", "Книжная страна");
    shop3 = new SHOP("shop3.txt", "Путешественник");
    warehouse1 = new WAREHOUSE("warehouse.txt");
    shop1->set_next(shop2);
    shop2->set_next(shop3);
```

```

shop3->set_next(warehouse1);

    cout << "Приветствуем Вас в нашем интернет магазине путеводителей!
Ознакомьтесь с нашим каталогом!" << endl;
    cout << "Если Вас что-то заинтересовало, вы можете сделать заказ прямо
тут!" << endl << endl;
    guide_catalog->show();
    cout << "Вы хотите сделать заказ? (Да/Нет)" << endl;
    cin >> choise;
    while (choise != "Нет" && choise != "нет")
    {
        cout << "\tВведите номер заинтересовавшего вас путеводителя: ";
        cin >> ch;
        if(ch > guide_catalog->get_amount())
        {
            cout << "К сожалению, такого товара нет в каталоге." << endl;
        }
        else
        {
            cout << "\tВы выбрали: " << guide_catalog->get_name(ch) << endl;
            temp = guide_catalog->get_vendor_code(ch);
            if(shop1->product_search(temp) == true)
            {
                guide_catalog->purchased(ch);
                cout << "\tБлагодарим за использование нашего сервиса." <<
endl;
            }
        }
        cout << "Хотите заказать что-то ещё? (Да/Нет)" << endl;
        cin >> choise;
    }
    cout << "До свидания!" << endl;
    return 0;
}

```

guide.h

```

#pragma once
#include <string>

using namespace std;

class GUIDE{
private:
    string name;
    string publishing_house;
    int year;
    int num_pages;
    bool illustrations;
    bool city_map;
    string binding;
    string format;
    string author;
    int quantity;
    string vendor_code;
public:
    void set_name(string temp);
    void set_publishing_house(string temp);
    void set_year(int temp);
    void set_num_pages(int temp);
    void set_illustrations(bool temp);
    void set_city_map(bool temp);

```

```

    void set_binding(string temp);
    void set_format(string temp);
    void set_author(string temp);
    void set_quantity(int temp);
    void set_vendor_code(string temp);

    string get_name();
    string get_publishing_house();
    int get_year();
    int get_num_pages();
    bool get_illustrations();
    bool get_city_map();
    string get_binding();
    string get_format();
    string get_author();
    int get_quantity();
    string get_vendor_code();

    void purchased(int order);
};

```

guide.cpp

```

#include "guide.h"

#include <string>

using namespace std;

void GUIDE::set_name(string temp)
{
    name = temp;
}
void GUIDE::set_publishing_house(string temp)
{
    publishing_house = temp;
}
void GUIDE::set_year(int temp)
{
    year = temp;
}
void GUIDE::set_num_pages(int temp)
{
    num_pages = temp;
}
void GUIDE::set_illustrations(bool temp)
{
    illustrations = temp;
}
void GUIDE::set_city_map(bool temp)
{
    city_map = temp;
}
void GUIDE::set_binding(string temp)
{
    binding = temp;
}
void GUIDE::set_format(string temp)
{
    format = temp;
}
void GUIDE::set_author(string temp)

```



```

{
    author = temp;
}
void GUIDE::set_quantity(int temp)
{
    quantity = temp;
}
void GUIDE::set_vendor_code(string temp)
{
    vendor_code = temp;
}

string GUIDE::get_name()
{
    return name;
}
string GUIDE::get_publishing_house()
{
    return publishing_house;
}
int GUIDE::get_year()
{
    return year;
}
int GUIDE::get_num_pages()
{
    return num_pages;
}
bool GUIDE::get_illustrations()
{
    return illustrations;
}
bool GUIDE::get_city_map()
{
    return city_map;
}
string GUIDE::get_binding()
{
    return binding;
}
string GUIDE::get_format()
{
    return format;
}
string GUIDE::get_author()
{
    return author;
}
int GUIDE::get_quantity()
{
    return quantity;
}
string GUIDE::get_vendor_code()
{
    return vendor_code;
}

void GUIDE::purchased(int order)
{
    quantity -= order;
}

```

catalog.h

```
#pragma once

#include "guide.h"

class CATALOG{
private:
    int amount;
    GUIDE *guides;
public:
    CATALOG();
    int get_amount();
    string get_vendor_code(int temp);
    string get_name(int temp);
    void purchased(int temp);
    void show();
};
```

catalog.cpp

```
#include "catalog.h"
#include "guide.h"

#include <cstdlib>
#include <stdio.h>
#include <iostream>
#include <fstream>

using namespace std;

CATALOG::CATALOG()
{
    ifstream file;
    string temp_string;
    int temp_int;
    bool temp_bool;

    file.open("site.txt");
    if (!file.is_open())
    {
        cout << "Файл site.txt не найден!" << endl;
        amount = 0;
        guides = NULL;
    }
    else
    {
        amount = 0;
        if (!file.eof())
        {
            getline(file, temp_string);
        }
        if (!file.eof())
        {
            guides = new GUIDE;
            amount++;
            getline(file, temp_string, '\t');
            guides->set_name(temp_string);
            getline(file, temp_string, '\t');
            guides->set_publishing_house(temp_string);
            file >> temp_int;
            guides->set_year(temp_int);
        }
    }
}
```

```

        file >> temp_int;
        guides->set_num_pages(temp_int);
        getline(file, temp_string, '\t');
        getline(file, temp_string, '\t');
        if (temp_string == "да")
            temp_bool = true;
        else
            temp_bool = false;
        guides->set_illustrations(temp_bool);
        getline(file, temp_string, '\t');
        if (temp_string == "да")
            temp_bool = true;
        else
            temp_bool = false;
        guides->set_city_map(temp_bool);
        getline(file, temp_string, '\t');
        guides->set_binding(temp_string);
        getline(file, temp_string, '\t');
        guides->set_format(temp_string);
        getline(file, temp_string, '\t');
        guides->set_author(temp_string);
        file >> temp_int;
        guides->set_quantity(temp_int);
        getline(file, temp_string, '\t');
        getline(file, temp_string);
        guides->set_vendor_code(temp_string);
    }
    while (!file.eof())
    {
        GUIDE *temp;
        temp = new GUIDE[amount];
        for (int i=0; i<amount; ++i)
            temp[i] = guides[i];
        amount++;
        guides = new GUIDE[amount];
        for (int i=0; i<(amount-1); ++i)
            guides[i] = temp[i];

        getline(file, temp_string, '\t');
        guides[amount-1].set_name(temp_string);
        getline(file, temp_string, '\t');
        guides[amount-1].set_publishing_house(temp_string);
        file >> temp_int;
        guides[amount-1].set_year(temp_int);
        file >> temp_int;
        guides[amount-1].set_num_pages(temp_int);
        getline(file, temp_string, '\t');
        getline(file, temp_string, '\t');
        if (temp_string == "да")
            temp_bool = true;
        else
            temp_bool = false;
        guides[amount-1].set_illustrations(temp_bool);
        getline(file, temp_string, '\t');
        if (temp_string == "да")
            temp_bool = true;
        else
            temp_bool = false;
        guides[amount-1].set_city_map(temp_bool);
        getline(file, temp_string, '\t');
        guides[amount-1].set_binding(temp_string);
        getline(file, temp_string, '\t');
    }
}

```

```

        guides[amount-1].set_format(temp_string);
        getline(file, temp_string, '\t');
        guides[amount-1].set_author(temp_string);
        file >> temp_int;
        guides[amount-1].set_quantity(temp_int);
        getline(file, temp_string, '\t');
        getline(file, temp_string);
        guides[amount-1].set_vendor_code(temp_string);
    }
    file.close();
}

int CATALOG::get_amount()
{
    return amount;
}

string CATALOG::get_vendor_code(int temp)
{
    return guides[temp-1].get_vendor_code();
}

string CATALOG::get_name(int temp)
{
    return guides[temp-1].get_name();
}

void CATALOG::purchased(int temp)
{
    guides[temp-1].purchased(1);
}

void CATALOG::show()
{
    if (guides == NULL)
    {
        cout << "Каталог пуст" << endl;
    }
    else
    {
        cout << " № |                название                | издательство |
год | стр. | фото | карта | переплет | формат   |   автор   | к-во" << endl;
        cout <<
"
        _____" << endl;
        for(int i=0; i<amount; ++i)
        {
            printf("%2d | %-35s | %-12s | %d | %3d |", i+1,
(guides[i].get_name()).c_str(), (guides[i].get_publishing_house()).c_str(),
guides[i].get_year(), guides[i].get_num_pages());
            if (guides[i].get_illustrations() == true)
                cout << " да |";
            else
                cout << " нет |";
            if (guides[i].get_city_map() == true)
                cout << " да |";
            else
                cout << " нет |";
            printf(" %-7s | %-7s | %-10s | %3d",
(guides[i].get_binding()).c_str(), (guides[i].get_format()).c_str(),
(guides[i].get_author()).c_str(), guides[i].get_quantity());

```

```

        cout << endl;
    }
}
return;
}

```

institution.h

```

#pragma once

#include <string>

using namespace std;

struct RECORD{
    string vendor_code;
    int quantity;
};

class INSTITUTION{
protected:
    int amount;
    RECORD *store;
    INSTITUTION *next;
public:
    void set_next(INSTITUTION *temp){next = temp;};
    virtual bool product_search(string temp_code) = 0;
};

```

shop.h

```

#pragma once

#include "institution.h"

using namespace std;

class SHOP: public INSTITUTION{
private:
    string name;
public:
    SHOP();
    SHOP(string file_name, string temp);
    bool product_search(string temp_code);
};

```

shop.cpp

```
#include "institution.h"
#include "shop.h"

#include <string>
#include <iostream>
#include <fstream>

using namespace std;

SHOP::SHOP()
{
    amount = 0;
    store = NULL;
    next = NULL;
    name = "Безымянный";
}

SHOP::SHOP(string file_name, string temp)
{
    ifstream file;
    string temp_string;
    int temp_int;

    name = temp;
    next = NULL;
    file.open(file_name.c_str());
    if (!file.is_open())
    {
        cout << "Файл " << file_name << " не найден!" << endl;
        amount = 0;
        store = NULL;
    }
    else
    {
        amount = 0;
        if (!file.eof())
        {
            getline(file, temp_string);
        }
        if (!file.eof())
        {
            store = new RECORD;
            amount++;
            getline(file, temp_string, ' ');
            store->vendor_code = temp_string;
            file >> temp_int;
            store->quantity = temp_int;
            getline(file, temp_string);
        }
        while (!file.eof())
        {
            RECORD *temp;
            temp = new RECORD[amount];
            for (int i=0; i<amount; ++i)
                temp[i] = store[i];
            amount++;
            store = new RECORD[amount];
            for (int i=0; i<(amount-1); ++i)
                store[i] = temp[i];
        }
    }
}
```

```

        getline(file, temp_string, ' ');
        store[amount-1].vendor_code = temp_string;
        file >> temp_int;
        store[amount-1].quantity = temp_int;
        getline(file, temp_string);
    }
    file.close();
}

bool SHOP::product_search(string temp_code)
{
    for(int i=0; i<amount; ++i)
    {
        if (store[i].vendor_code == temp_code && store[i].quantity > 0)
        {
            store[i].quantity--;
            cout << "\tОформлена доставка из магазина " << name << "." << endl;
            return true;
        }
    }
    if(next != NULL)
    {
        if (next->product_search(temp_code) == true)
            return true;
    }
    return false;
}

```

warehouse.h

```

#pragma once

#include "institution.h"

using namespace std;

class WAREHOUSE: public INSTITUTION{
public:
    WAREHOUSE();
    WAREHOUSE(string file_name);
    bool product_search(string temp_code);
};

```

warehouse.cpp

```
#include "institution.h"
#include "warehouse.h"

#include <string>
#include <iostream>
#include <fstream>

using namespace std;

WAREHOUSE::WAREHOUSE()
{
    amount = 0;
    store = NULL;
    next = NULL;
}

WAREHOUSE::WAREHOUSE(string file_name)
{
    ifstream file;
    string temp_string;
    int temp_int;

    next = NULL;
    file.open(file_name.c_str());
    if (!file.is_open())
    {
        cout << "Файл " << file_name << " не найден!" << endl;
        amount = 0;
        store = NULL;
    }
    else
    {
        amount = 0;
        if (!file.eof())
        {
            getline(file, temp_string);
        }
        if (!file.eof())
        {
            store = new RECORD;
            amount++;
            getline(file, temp_string, ' ');
            store->vendor_code = temp_string;
            file >> temp_int;
            store->quantity = temp_int;
            getline(file, temp_string);
        }
        while (!file.eof())
        {
            RECORD *temp;
            temp = new RECORD[amount];
            for (int i=0; i<amount; ++i)
                temp[i] = store[i];
            amount++;
            store = new RECORD[amount];
            for (int i=0; i<(amount-1); ++i)
                store[i] = temp[i];

            getline(file, temp_string, ' ');
            store[amount-1].vendor_code = temp_string;
```



```

        file >> temp_int;
        store[amount-1].quantity = temp_int;
        getline(file, temp_string);
    }
    file.close();
}

bool WAREHOUSE::product_search(string temp_code)
{
    string choise;
    for(int i=0; i<amount; ++i)
    {
        if (store[i].vendor_code == temp_code && store[i].quantity > 0)
        {
            cout << "\tТовар в наличии на складе. Забронировать товар для
самовывоза? (Да/Нет)" << endl << "\t";
            cin >> choise;
            if (choise == "Да" || choise == "да")
            {
                store[i].quantity--;
                cout << "\tТовар забронирован." << endl;
                return true;
            }
            else
            {
                cout << "\tЗаказ отменён." << endl;
                return false;
            }
        }
    }
    if(next != NULL)
    {
        if (next->product_search(temp_code) == true)
            return true;
    }
    cout << "\tИзвините, товара нет в наличии." << endl;
    return false;
}

```

3. Список использованной литературы

- 1) Керниган Б., Ритчи Д. – Язык программирования Си. Учебное пособие. – 3-е изд., испр. — СПб.: «Невский Диалект», 2001г. – 188 с.
- 2) Роберт Лафоре. Объектно-ориентированное программирование в C++
- 3) <https://metanit.com/sharp/patterns/3.7.php>
- 4) <https://refactoring.guru/ru/design-patterns/chain-of-responsibility>
- 5) <https://habr.com/ru/post/113995/>

4. Примеры работы программы:

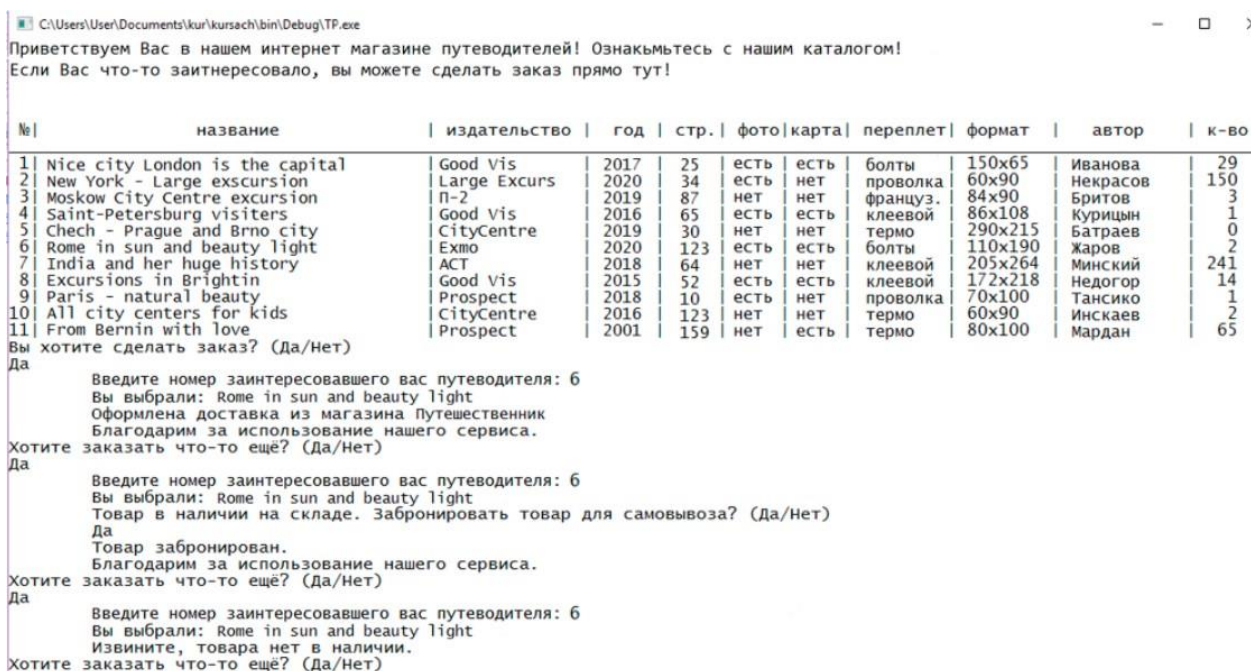


Приветствуем Вас в нашем интернет магазине путеводителей! Ознакомьтесь с нашим каталогом!
Если Вас что-то заинтересовало, вы можете сделать заказ прямо тут!

№	название	издательство	год	стр.	фото	карта	переплет	формат	автор	к-во
1	Nice city London is the capital	Good Vis	2017	25	есть	есть	болты	150x65	Иванова	29
2	New York - Large excursion	Large Excurs	2020	34	есть	нет	проволка	60x90	Некрасов	150
3	Moscow City Centre excursion	П-2	2019	87	нет	нет	француз.	84x90	Бритов	3
4	Saint-Petersburg visitors	Good Vis	2016	65	есть	есть	клеевой	86x108	Курицын	1
5	Czech - Prague and Brno city	CityCentre	2019	30	нет	нет	термо	290x215	Батраев	0
6	Rome in sun and beauty light	Exmo	2020	123	есть	есть	болты	110x190	Жаров	2
7	India and her huge history	ACT	2018	64	нет	нет	клеевой	205x264	Минский	241
8	Excursions in Brightin	Good Vis	2015	52	есть	есть	клеевой	172x218	Недогор	14
9	Paris - natural beauty	Prospect	2018	10	есть	нет	проволка	70x100	Тансико	1
10	All city centers for kids	CityCentre	2016	123	нет	нет	термо	60x90	Инскаев	2
11	From Bernin with love	Prospect	2001	159	нет	есть	термо	80x100	Мардан	65

Вы хотите сделать заказ? (Да/Нет)

Рисунок 1 – Запуск программы



Приветствуем Вас в нашем интернет магазине путеводителей! Ознакомьтесь с нашим каталогом!
Если Вас что-то заинтересовало, вы можете сделать заказ прямо тут!

№	название	издательство	год	стр.	фото	карта	переплет	формат	автор	к-во
1	Nice city London is the capital	Good Vis	2017	25	есть	есть	болты	150x65	Иванова	29
2	New York - Large excursion	Large Excurs	2020	34	есть	нет	проволка	60x90	Некрасов	150
3	Moscow City Centre excursion	П-2	2019	87	нет	нет	француз.	84x90	Бритов	3
4	Saint-Petersburg visitors	Good Vis	2016	65	есть	есть	клеевой	86x108	Курицын	1
5	Czech - Prague and Brno city	CityCentre	2019	30	нет	нет	термо	290x215	Батраев	0
6	Rome in sun and beauty light	Exmo	2020	123	есть	есть	болты	110x190	Жаров	2
7	India and her huge history	ACT	2018	64	нет	нет	клеевой	205x264	Минский	241
8	Excursions in Brightin	Good Vis	2015	52	есть	есть	клеевой	172x218	Недогор	14
9	Paris - natural beauty	Prospect	2018	10	есть	нет	проволка	70x100	Тансико	1
10	All city centers for kids	CityCentre	2016	123	нет	нет	термо	60x90	Инскаев	2
11	From Bernin with love	Prospect	2001	159	нет	есть	термо	80x100	Мардан	65

Вы хотите сделать заказ? (Да/Нет)
Да
Введите номер заинтересовавшего вас путеводителя: 6
Вы выбрали: Rome in sun and beauty light
Оформлена доставка из магазина Путешественник
Благодарим за использование нашего сервиса.
Хотите заказать что-то еще? (Да/Нет)
Да
Введите номер заинтересовавшего вас путеводителя: 6
Вы выбрали: Rome in sun and beauty light
Товар в наличии на складе. Забронировать товар для самовывоза? (Да/Нет)
Да
Товар забронирован.
Благодарим за использование нашего сервиса.
Хотите заказать что-то еще? (Да/Нет)
Да
Введите номер заинтересовавшего вас путеводителя: 6
Вы выбрали: Rome in sun and beauty light
Извините, товара нет в наличии.
Хотите заказать что-то еще? (Да/Нет)

Рисунок 2 – Визуализация цепочки обязанностей

site.txt — Блокнот										
Файл Правка Формат Вид Справка										
название, издательство, год, число страниц, наличие фотографий, наличие карты, переплет, формат, автор, количество, артикул										
Nice city London is the capital	Good Vis	2017	25	есть	есть	болты	150x65	Иванова	29	12076
New York - Large excursion	Large Excurs	2020	34	есть	нет	проволка	60x90	Некрасов	150	18723
Moscow City Centre excursion	п-2	2019	87	нет	нет	француз.	84x90	Бритов	3	20154
Saint-Petersburg visitors	Good Vis	2016	65	есть	есть	клеевой	86x108	Курицын	1	19324
Czech - Prague and Brno city	CityCentre	2019	30	нет	нет	термо	290x215	Батраев	0	54621
Rome in sun and beauty light	Exmo	2020	123	есть	есть	болты	110x190	Жаров	2	34097
India and her huge history	ACT	2018	64	нет	нет	клеевой	205x264	Минский	241	12097
Excursions in Brighton	Good Vis	2015	52	есть	есть	клеевой	172x218	Недогор	14	42683
Paris - natural beauty	Prospect	2018	10	есть	нет	проволка	70x100	Тансико	1	24013
All city centers for kids	CityCentre	2016	123	нет	нет	термо	60x90	Инскаев	2	53025
From Bernin with love	Prospect	2001	159	нет	есть	термо	80x100	Мардан	65	23154

Рисунок 3 – Файл с каталогом и информацией о товарах.

C:\Users\User\Documents\kur\kursach\bin\Debug\TP.exe

Приветствуем Вас в нашем интернет магазине путеводителей! Ознакомьтесь с нашим каталогом!
Если Вас что-то заинтересовало, вы можете сделать заказ прямо тут!

№	название	издательство	год	стр.	фото	карта	переплет	формат	автор	к-во
1	Nice city London is the capital	Good Vis	2017	25	есть	есть	болты	150x65	Иванова	29
2	New York - Large excursion	Large Excurs	2020	34	есть	нет	проволка	60x90	Некрасов	150
3	Moscow City Centre excursion	п-2	2019	87	нет	нет	француз.	84x90	Бритов	3
4	Saint-Petersburg visitors	Good Vis	2016	65	есть	есть	клеевой	86x108	Курицын	1
5	Czech - Prague and Brno city	CityCentre	2019	30	нет	нет	термо	290x215	Батраев	0
6	Rome in sun and beauty light	Exmo	2020	123	есть	есть	болты	110x190	Жаров	2
7	India and her huge history	ACT	2018	64	нет	нет	клеевой	205x264	Минский	241
8	Excursions in Brighton	Good Vis	2015	52	есть	есть	клеевой	172x218	Недогор	14
9	Paris - natural beauty	Prospect	2018	10	есть	нет	проволка	70x100	Тансико	1
10	All city centers for kids	CityCentre	2016	123	нет	нет	термо	60x90	Инскаев	2
11	From Bernin with love	Prospect	2001	159	нет	есть	термо	80x100	Мардан	65

Вы хотите сделать заказ? (Да/Нет)
Да

Введите номер заинтересовавшего вас путеводителя: 9
Вы выбрали: Paris - natural beauty
Товар в наличии на складе. Забронировать товар для самовывоза? (Да/Нет)
Нет
Заказ отменён.

Хотите заказать что-то ещё? (Да/Нет)
да

Введите номер заинтересовавшего вас путеводителя: 9
Вы выбрали: Paris - natural beauty
Товар в наличии на складе. Забронировать товар для самовывоза? (Да/Нет)
да
Товар забронирован.
Благодарим за использование нашего сервиса.
Хотите заказать что-то ещё? (Да/Нет)

Рисунок 4 – Бронирование товара со склада