МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Р.Е. АЛЕКСЕЕВА

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ

ТЕХНОЛОГИЙ

**КУРСОВАЯ РАБОТА**

**“Информационно-поисковая система”**

Выполнил:

студент группы 10-ТР А.А. Рябов

Проверил:

доц. кафедры ИРС С.Б. Сидоров

Нижний Новгород

2011

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ.......................................................................................................................................... 3

ПОСТАНОВКА ЗАДАЧИ...................................................................................................................... 4

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ....................................................................................................... 5

РУКОВОДСТВО ПРОГРАММИСТА..................................................................................................... 9

Структура программы.................................................................................................................. 9

Структуры данных 10

Алгоритм 10

ЗАКЛЮЧЕНИЕ................................................................................................................................... 12

СПИСОК ЛИТЕРАТУРЫ..................................................................................................................... 13

ПРИЛОЖЕНИЕ.................................................................................................................................. 14

Основная программа 14

Модуль “ Работа с файлом ” 14

Модуль “Интерфейс пользователя”......................................................................................... 19

Модуль “ Команды ”................................................................................................................... 22

Модуль “ Редактор продукции ”................................................................................................ 26

Модуль “ База ”........................................................................................................................... 32

**ВВЕДЕНИЕ**

В работе рассматривается решение задачи разработки прикладной программы информационно-поисковой системы.

Одним из способов преодоления сложности данной задачи является ис­пользование технологии разработки программ «сверху - вниз». Этот подход поз­воляет сначала задачу разбить на более простые подзадачи и составить основную программу. Вместо тех процедур, которые еще не реализованы, пишутся «заглушки» – процедуры и функции, которые ничего не делают, но имеют заданный список параметров. Затем каждая подзадача разбивается на еще более мелкие подзадачи и так далее, пока все процедуры не будут реализованы. Одним из достоинств этого метода является меньшая вероятность принципиальной ошибки. Используя этот метод структура программы получается простой и понятной. Также существенным мо­ментом технологии «сверху - вниз» является её акцент на повторное использова­ние уже существующих модулей, а не разработку их «с нуля».

К недостаткам может быть отнесена потребность реализовывать несколько похожих процедур или функций, вместо которых можно было бы обойтись одной.

В начале основной части отчёта приводится точная формализованная по­становка задачи с указанием полного набора операций, выполнение которых должна обеспечивать прикладная программа.

В руководстве пользователя раскрывается назначение программы, её воз­можности и выполняемые операции. Подробно объясняются правила пользова­ния программой и приводятся конкретные примеры диалога с пользователем.

В руководстве программиста рассматриваются вопросы внутренней организации программы, в том числе её состав модулей и их взаимодействие. Кро­ме этого описываются используемые структуры данных и наиболее важные и интересные с точки зрения их реализации алгоритмы.

В заключении делаются выводы о полноте решения поставленной задачи, а также приводится ряд соображений по возможным направлениям доработки полученной прикладной программы.

Также приведена техническая информация, включающая листинги всех модулей программы.

**ПОСТАНОВКА ЗАДАЧИ**

Требуется разработать справочную систему "Товары", содержащую сведения

об имеющихся в наличии товарах в магазине. Каждый вид товара описывается

следующим образом:

- название;

- количество (в единицах - кг, штук и т.д.);

- цена за единицу;

- срок реализации;

- поставщики.

Пользователь системы должен иметь возможность выполнять следующие операции:

- добавления нового товара;

- удаления товара;

- корректировка товара;

- поиск товаров по значению некоторых атрибутов;

- вывод информации по заданному названию товара;

- выдать список товаров, у которых срок реализации истекает ранее указанной даты;

- сформировать список поставщиков указанной продукции;

- принять и выполнить заказ на приобретение указанного товара с заданным

количество единиц. (Исполнения заказа означает внесение изменений в базу данных о товаре).

Одним из требований к разрабатываемой программной системе является хранение таблицы с исходными данными в файле. Все изменения, вносимые в базу данных, **не должны теряться** при завершении работы с программой, то есть необходимо обновление содержимого файла при изменении базы данных.

Взаимодействие пользователя с программой осуществляется через меню.

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

Данная программа предназначена для содержания сведений об имеющихся в наличии товарах в магазине. Работа с программой представляет из себя интерактивный диалог. Программа выдаёт меню с доступными задачами, пользователь выбирает нужный пункт, затем вводит необходимые данные и получает результат. После выполнения задачи можно ее продолжить или вернуться на один шаг выше.

Запуск программы на выполнение осуществляется либо набором в ко­мандной строке имени программы **tovar.exe** с последующим нажатием на кла­вишу «**Enter**», либо каким-то другим способом, зависящем от конкретной опе­рационной системы.

После запуска программы на экране монитора появляется краткое информационное сообщение о программе следующего вида:

*Программа «справочная система товары»*

*Автор: Рябов А.А.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Программа является интерактивным консольным приложением. Весь диалог с пользователем осуществляется в текстовом режиме через главное меню 1.0 , главное меню2.0 и нескольких второстепенных меню, структуры которых приведены ниже.

Главные меню появляются автоматически после выдачи сообщения о программе.

Причем главное меню 1.0 выводится, если в папке данного приложения файл, содержащий информацию о товарах отсутствует или он пуст. Оно имеет следующую структуру:

*1 > добавление товара в справочную систему*

*0 > выход*

А если файл не пуст, то выводится главное меню2.0 , которое имеет структуру вида:

*1 > вывод информации по заданному названию товара*

*2 > вывод списка товаров с истекшим сроком реализации*

*3 > формирование списка поставщиков указанной продукции*

*4 > выполнение заказа на приобретение товара*

*5> удаление товара*

*6 > добавление товара*

*7 > корректировка товара*

*8 > поиск товара*

*0 > выход*

Второстепенное меню 1.0 вида:

*1 > ПОВТОРИТЬ*

*0 > НАЗАД*

*? >*

позволяет при вводе "**1**" повторить выбранный ранее пункт в главных или второстепенных меню,

при вводе "**0**" выполнить шаг назад.

После отображения одного из главных меню пользователю предлагается указать номер тре­буемого действия:

*? >*

При вводе недопустимого номера одно из главных меню с последующим запросом отоб­ражается заново.

Написанное ниже справедливо, если выводится главное меню2.0 . В противном случае необходимо перейти к описанию пункта №6 (6 > добавление товара), проигнорировав 5 идущих перед ним.

При выборе пункта №1 главного меню2.0 выполняется операция вывода информации по заданному названию товара. При этом на экране монитора появляется запрос вида:

*Введите название товара*

После указания пользователем товара, если он найден осуществляется вывод по нему информации, иначе выводится сообщение, что товар не найден . Затем выводится второстепенное меню 1.0 (см. выше).

При выборе пункта №2 главного меню2.0 пользователю предлагается указать срок реализации товара, при этом выполняются следующие запросы:

*Введите число месяца (от 1 до 31)*

*Введите месяц (от 1 до 12)*

*Введите год*

После этого выводятся товары с истекшим сроком реализации. Затем выводится второстепенное меню 1.0 (см. выше).

При выборе пункта№3 главного меню2.0 выполняется операция формирования поставщиков по заданному названию товара. При этом на экране монитора появляется запрос вида:

*Введите название товара*

После указания пользователем товара, если он найден осуществляется вывод по нему поставщиков, иначе выводится сообщение, что товар не найден . Затем выводится второстепенное меню 1.0 (см. выше).

При выборе пункта №4 главного меню2.0 пользователь может заказать указанный товар с заданным его количество, при этом выполняется следующий запрос:

*Введите название товара*

После указания пользователем товара, если он не найден выводится сообщение, что товар не найден, иначе выводится сообщение

*Введите количество заказываемого товара (например: 12 кг или 15 шт )*

Если указанное количество товара есть в магазине, то выводится сообщение о название товара и о его количестве заказанного вами, иначе выводится сообщение вида:

*Такого количества нет в наличии*

Затем выводится второстепенное меню 1.0 (см. выше).

При выборе пункта №5 главного меню2.0 пользователь может выполнить удаления указанного товара, при этом выполняется следующий запрос:

*Введите название товара*

После указания пользователем товара, если он не найден выводится сообщение, что товар не найден, иначе выводится сообщение

*товар успешно удален*

Затем выводится второстепенное меню 1.0 (см. выше).

При выборе пункта №6 главного меню2.0 пользователь может выполнить добавления указанного товара, при этом выводится сообщения

*Введите:*

и выполняются следующие запросы:

1. *название товара*
2. *количество товара (например: 12 кг или 15 шт и д.р.)*
3. *цену товара за единицу (например: 15 $ или 180 руб и д.р.)*
4. *срок реализации товара*

*введите число месяца (от 1 до 31)*

*введите месяц (от 1 до 12)*

*введите год*

1. *поставщика товара*

После выводится второстепенное меню 2.0 вида:

*1 > ЕЩЕ ПОСТАВЩИК*

*0 > ХВАТИТ*

*? >*

Которое позволяет при вводе "**1**" ввести еще поставщика, указанного товара.

При вводе "**0**" выводится второстепенное меню 1.0 (см. выше).

При выборе пункта №7 главного меню2.0 пользователю предоставляется возможность от корректировать добавленный ранее товар, при этом выполняется запрос вида:

*Введите название товара*

Если товар не был найден осуществляется вывод сообщения, что товар не найден.

Иначе выводится второстепенного меню3.0 вида:

*корректировка:*

*1 > имени товара*

*2 > количества товара*

*3 > цены товара за единицу*

*4 > срока реализации товара*

*5 > поставщиков товара*

*0 > назад*

*? >*

При вводе пункта от 1 до 5 второстепенного меню3.0 производятся те же операции, что и пронумерованные соответственно в пункте №6 главного меню2.0

После выводится второстепенное меню 1.0 (см. выше).

При выборе пункта №8 главного меню2.0 пользователь может отыскать необходимый товар

через второстепенное меню4.0 вида:

1 > поиск товара по его количеству

2 > поиск товара по его цене

3 > поиск товара по его поставщику

4 > поиск товара по всем выше перечисленным пунктам

0 > назад

? >

При вводе 1, 2 или 3 поиск ведется по количеству, цене или по поставщику товара соответственно.

При вводе 4 поиск ведется сразу по всем 3, выше описанным пунктам.

При выборе пункта №0 главного меню2.0 производится выход из программы.

**РУКОВОДСТВО ПРОГРАММИСТА**

***Структура программы***

Прикладная программа разработана с использованием принципов модульного

программирования. Она является совокупностью взаимодействующих модулей. Модули

объединяются в единую программу путем компоновки соответствующих объектных файлов.

Структура программы представлена ниже.

"Приложение"

"Работа с файлом"

"Интерфейс пользователя"

"Команды"

"Редактор продукции"

"База"

Программа состоит из 6 модулей назначение каждого из которых приведено ниже:

1. main.c – головной модуль;
2. work\_file.h, work\_file.c - здесь находятся функции отвечающие за работу программы с файлом;
3. user\_interface.h, user\_interface.c – здесь находятся функции, которые отвечают за “общение” с пользователем;
4. edit\_product.h, edit\_product.c – здесь собраны функции, непосредственно отвечающие за редактирование продукции, имеющейся в "Информационно-поисковой системе";
5. comandy.h, comandy.c - здесь собраны функции, с помощью которых можно найти, заказать необходимый товар, просмотреть информацию о нём;
6. base.h, base.c - здесь находятся функции, отвечающие за ввод-вывод срока реализации, поставщиков товара, а также функция проверки правильного ввода срока реализации.

***Структуры данных***

Для описания товара в "Информационно-поисковой системе" используются 3 типа данных. Это структуры kp\_tovar, post\_tovar и TOVAR. Первая из них предназначена для описания количества и цены товара, вторая для описания поставщиков товара, а последняя же структура описывает товар в целом.

Выше перечисленные структуры приведены ниже:

struct kp\_tovar //структура для количества и цены товара

{

float kp; //значени

char A[20]; //единицы измерения

};

struct post\_tovar // структура для поставщиков

{

int number; //количество поставщиков товара, max 80

char post[80][80]; // поставщики

};

struct TOVAR

{

char name[80]; //название товара

struct kp\_tovar kol; //количество товара

struct kp\_tovar price;//цена за единицу

struct tm date; //срок реализации

struct post\_tovar sup; //поставщики товара

};

***Алгоритм***

Алгоритм данной программы прост и не требует подробного объяснения.

Можно лишь сказать несколько слов об алгоритме сохранения данных о товарах в файл. Выполнил какую-либо команду в программе и вернувшись в главное меню2.0 , данные сразу же сортируются по имени товара в алфавитном порядке, а затем сохраняются в файл в виде структур данных типа struct TOVAR.

**ЗАКЛЮЧЕНИЕ**

В данной работе задача разработки информационно-поисковой системы была решена с использованием технологии «сверху - вниз». На основании проведённой отладки и испытаний с помощью контрольных примеров можно сделать вывод, что полученная программа решает поставленную задачу правильно и в полном объёме.

В программе диалог с пользователем осуществляется в консольном режиме. Одним из

возможных направлений доработки данной программы является реализация более наглядного

графического интерфейса пользователя. В этом случае изменится только один модуль

программы, отвечающий за организацию диалога с пользователем «Интерфейс пользователя».

Это можно утверждать, поскольку интерфейсная часть этого модуля в части набора

предоставляемых операций и их прототипов не изменится. Изменениям подвергнется только

реализация данных операций.

Другим перспективным направлением доработки информационно-поисковой системы является возможность сохранения данных о товарах не в двоичный файл, а в текстовый для его наглядности и редактирования. При этом доработать придется только модуль «Работа с файлом», изменив соответствующие функции.

Так как для всех рассмотренных случаев возможные доработки имеют локальный характер и не требуют изменения общей структуры программы, то можно сделать вывод о

гибкости полученного программного обеспечения и, следовательно, об успешно проведенной декомпозиции задачи.

**СПИСОК ЛИТЕРАТУРЫ**

1. Подбельский В. В., Фомин С. С. Программирование на языке Си: Учеб. пособие. - 2-е доп. изд. - М.: Финансы и статистика, 2003. - 600 с.: ил.

2. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования C, 2е издание. [пер. с анг.] / Б.У. Керниган, Д.М. Ритчи – М.: Вильямс, 2006.

3. Орлов, С.А. Технологии разработки программного обеспечения. учеб. Пособие. 2е изд./ С.А. Орлов, СПб.: Питер, 2003. 480с.: ил.

**ПРИЛОЖЕНИЕ**

***Основная программа***

**//main.cpp**

**//Головной модуль**

*#include "work\_file.h"*

*int main(void)*

*{*

*call\_func();*

*return 0;*

*}*

***Модуль “*** **Работа с файлом *”***

**//****work\_file.h**

**//Интерфейс модуля *“*** **Работа с файлом *”***

*#include <stdio.h>*

*#ifndef work\_file\_h*

*#define work\_file\_h*

*int kolstrok(FILE \*fp);*

*int KOL\_STR(void);*

*int menu\_2(struct TOVAR \*M, int N, int k);*

*int menu\_1(struct TOVAR \*M, int N, int k);*

*int panel(struct TOVAR \*M, int N, int k);*

*int sort\_tovar(struct TOVAR \*M, unsigned int n );*

*int fwrite\_tovar(struct TOVAR \*M, int B);*

*int call\_func(void);*

*#endif*

**//****work\_file.cpp**

**//Реализация модуля *“*** **Работа с файлом *”***

*#include <stdio.h>*

*#include <string.h>*

*#include "edit\_product.h"*

*#include "comandy.h"*

*#include "user\_interface.h"*

*#include "base.h"*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int kolstrok(FILE \*fp) //функция для подсчета структур в файле программы*

*{*

*struct TOVAR G;*

*int kolst=0; //kol-количество строк*

*while (fread ( &G, sizeof(struct TOVAR),1 , fp )!=0)*

*kolst++;*

*return kolst;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int KOL\_STR(void) //функция открытия и закрытия файла для подсчета в нем структур типа struct TOVAR*

*{*

*int N;*

*FILE \*f;*

*f=fopen("tovar","rb");*

*if (f == NULL)*

*N=0;*

*else*

*N=kolstrok(f);*

*fclose(f);*

*return N;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int menu\_2(struct TOVAR \*M, int N, int k)*

*{*

*int a=1;*

*int B;*

*B=N;*

*switch(k)*

*{*

*case 1 :*

*{*

*Dobavit\_Element(M, B);*

*break;*

*}*

*case 2 :*

*{*

*InfoTovar(M, N);*

*break;*

*}*

*case 3 :*

*{*

*SrokRealiz(M, N);*

*break;*

*}*

*case 4 :*

*{*

*Postavshik(M, N);*

*break;*

*}*

*case 5 :*

*{*

*Zakaz(M, N);*

*break;*

*}*

*case 6 :*

*{*

*Delete\_Element(M, B);*

*break;*

*}*

*case 7 :*

*{*

*Korrected\_tovar(M, N);*

*break;*

*}*

*case 8 :*

*{*

*Poisk\_tovar(M, N);*

*break;*

*}*

*default: break;*

*}*

*return B;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int menu\_1(struct TOVAR \*M, int N, int k)*

*{*

*int a=1;*

*int B;*

*B=N;*

*switch(k)*

*{*

*case 1 :*

*{*

*Dobavit\_Element(M, B);*

*break;*

*}*

*default: break;*

*}*

*return B;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int panel(struct TOVAR \*M, int N, int k)*

*{*

*int B;*

*if(N!=0)*

*B=menu\_2(M, N, k);*

*else*

*B=menu\_1(M, N, k);*

*return B;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int sort\_tovar (struct TOVAR \*M, unsigned int n )*

*{*

*struct TOVAR p;*

*int i, j;*

*if (n == 0)*

*return 0;*

*for ( i = 0; i < n-1; i ++ )*

*for ( j = n-1; j > i; j -- )*

*if ( strcmp(M[j-1].name,M[j].name) > 0 )*

*{*

*p = M[j];*

*M[j] = M[j-1]; // перестановка товара*

*M[j-1] = p;*

*}*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int fwrite\_tovar(struct TOVAR \*M, int B)*

*{*

*FILE \*fp;*

*fp=fopen("tovar","wb");*

*if (fp == NULL)*

*{*

*printf("Ошибка открытия файла\n");*

*return 0;*

*}*

*fwrite(M, sizeof(struct TOVAR), B, fp);*

*fclose(fp);*

*return 1;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int call\_func(void) //вызов ф-ий*

*{*

*int N, k, B, d;*

*struct TOVAR \*M;*

*printf\_info(); //вывод инфо о программе*

*do*

*{*

*N=KOL\_STR();*

*M=new struct TOVAR[N+11];//выделяет динамическую память*

*if (M == NULL)*

*{*

*printf("Ошибка выделения памяти\n");*

*break;*

*}*

*k=menu\_f(M, N); //вывод продукции и меню*

*B=panel(M, N, k); //команды управления программой*

*sort\_tovar(M, B); //сортирует товар в алфавитном порядке*

*d=fwrite\_tovar(M, B);*

*delete M; //очистить память*

*}*

*while(k != 0 && d != 0);*

*return 0;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

***Модуль “*Интерфейс пользователя*”***

**//****user\_interface.h**

**//Интерфейс модуля *“*** **Интерфейс пользователя *”***

*#ifndef comandy\_h*

*#define comandy\_h*

*void printf\_info(void);*

*void Menu\_Printf(struct TOVAR \*M, int N);*

*int menu\_f(struct TOVAR \*M, int N);*

*void Delete\_Element(struct TOVAR \*M, int &N);*

*void Dobavit\_Element(struct TOVAR \*M, int &B);*

*#endif*

**//****user\_interface.cpp**

**//Реализация модуля *“*** **Интерфейс пользователя *”***

*#include "base.h"*

*#include <stdio.h>*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*static char\* menu[]=*

*{*

*" 1 > добавление товара",*

*" 2 > вывод информации по заданному названию товара",*

*" 3 > вывод списка товаров с истекшим сроком реализации",*

*" 4 > формирование списка поставщиков указанной продукции",*

*" 5 > выполнение заказа на приобретение товара",*

*" 6 > удаление товара",*

*" 7 > корректировка товара",*

*" 8 > поиск товара",*

*" 0 > выход",*

*" ? > "*

*};*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void printf\_info(void)*

*{*

*printf("Программа \"справочная система товары\"\n");*

*printf("Автор: Рябов А.А.\n");*

*printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void Menu\_Printf(struct TOVAR \*M, int N) //функция вывода имен товара*

*{*

*int i;*

*for(i=0; i<N; i++)*

*printf(" | %s;\n", M[i].name);*

*printf("\n");*

*return ;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int menu\_f(struct TOVAR \*M, int N)*

*{*

*int i, number;*

*FILE \*fp;*

*do*

*{*

*printf("\nКоличество продукции в магазине %d\n",N);*

*if(N==0)*

*printf("%s\n", menu[0]);*

*else*

*{*

*fp=fopen("tovar","rb");*

*fread(M ,sizeof(struct TOVAR), N, fp);*

*fclose(fp);*

*printf(" Список продукции нашего магазина\n\n");*

*Menu\_Printf(M,N);*

*for(i=0; i<8; i++)*

*printf("%s\n",menu[i]);*

*}*

*printf("%s\n", menu[8]);*

*printf("%s\a", menu[9]);*

*fflush(stdin);*

*scanf("%d",&number);*

*}*

*while(number < 0 || number > 8);*

*return number;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void Delete\_Element(struct TOVAR \*M, int &N)*

*{*

*int i;*

*while(N != 0)*

*{*

*i=Soarch\_Element(M, N);*

*if(i >= 0)*

*{*

*M[i] = M[N-1];*

*N --;*

*puts(" товар успешно удален!");*

*}*

*if(N != 0 && Vyxod() == 0 )*

*break;*

*}*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void Dobavit\_Element(struct TOVAR \*M, int &B)*

*{*

*int j, N;*

*N=B;*

*do*

*{*

*if((B-N) == 11)*

*{*

*printf(" невозможно добавить товар!\n Попробуйте повторить\n");*

*break;*

*}*

*M+=B;*

*B++;*

*puts("Введите:");*

*puts(" название товара");*

*fflush(stdin);*

*gets(M->name);*

*puts(" количество товара (например: 12 кг или 15 шт)");*

*scanf\_kptovar(&M->kol);*

*printf(" цену товара за %s (например: 15 $ или 180 руб)\n", M->kol.A);*

*scanf\_kptovar(&M->price);*

*puts(" срок реализации товара");*

*scanf\_date(&M->date);*

*puts(" поставщика товара");*

*scanf\_post\_tovar(&M->sup);*

*puts(" товар успешно добавлен!");*

*j=Vyxod();*

*}*

*while(j == 1);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

***Модуль “*** **Команды *”***

**//****comandy.h**

**//Интерфейс модуля *“*** **Команды *”***

*#ifndef comandy\_h*

*#define comandy\_h*

*void InfoTovar(struct TOVAR \*M, int N);*

*void Postavshik(struct TOVAR \*M, int N);*

*void Zakaz(struct TOVAR \*M, int N);*

*void SrokRealiz(struct TOVAR \*M, int N);*

*void search\_switch(struct TOVAR \*M, int N, int k);*

*void Poisk\_tovar(struct TOVAR \*M, int B);*

*#endif*

**//****comandy.cpp**

**//Реализация модуля *“*** **Команды *”***

*#include <stdio.h>*

*#include <string.h>*

*#include <time.h>*

*#include "base.h"*

*#include "edit\_product.h"*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*static char\* poisk[] =*

*{*

*" 1 > поиск товара по его количеству",*

*" 2 > поиск товара по его цене",*

*" 3 > поиск товара по его поставщику",*

*" 4 > поиск товара по всем выше перечисленным пунктам",*

*" 0 > назад",*

*" ? > "*

*};*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void InfoTovar(struct TOVAR \*M, int N)*

*{*

*int m;*

*do*

*{*

*m=Soarch\_Element(M, N);*

*if(m >= 0)*

*printf\_name\_tovar(M, m);*

*m=Vyxod();*

*}*

*while(m == 1);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void Postavshik(struct TOVAR \*M, int N) //функция формирования поставщиков товара*

*{*

*int i;*

*do*

*{*

*i = Soarch\_Element(M, N);*

*if (i >= 0)*

*printf\_post\_tovar(&M[i].sup);*

*i=Vyxod();*

*}*

*while(i == 1);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void Zakaz(struct TOVAR \*M, int N)*

*{*

*int i;*

*struct kp\_tovar f;*

*do*

*{*

*i = Soarch\_Element(M, N);*

*if(i >= 0)*

*{*

*puts(" \nВведите количество заказываемого товара (например: 12 кг или 15 шт)\n");*

*scanf\_kptovar(&f);*

*if (strcmp(M[i].kol.A, f.A) != 0 || M[i].kol.kp < f.kp)*

*puts("\n Такого количества нет в наличии\n");*

*else*

*{*

*M[i].kol.kp -= f.kp;*

*printf("вы заказали\n");*

*printf(" %s", M[i].name);*

*printf(" в количестве\n");*

*printf\_kptovar(&f);*

*}*

*}*

*else*

*printf("товар не найден!\n");*

*i = Vyxod();*

*}*

*while(i == 1);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void SrokRealiz(struct TOVAR \*M, int N)*

*{*

*int i;*

*struct tm t;*

*time\_t t1, t2; //Значение time\_t количество секунд прошедших с некоторой конкретной реализации базы времени.*

*do*

*{*

*scanf\_date(&t);*

*t1=mktime(&t); //mktime преобразует местное время, заданное структурой \*tp в тип данных time\_t*

*for (i=0; i<N; i++)*

*{*

*t2=mktime(&M->date);*

*if (0 > difftime(t2, t1)) //difftime(t2, t1) выдает pазницу (в секундах) между t2 и t1, типа double.*

*printf("\n%s",M->name);*

*M++;*

*}*

*i=Vyxod();*

*}*

*while(i == 1);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void search\_switch(struct TOVAR \*M, int N, int k)*

*{*

*switch(k)*

*{*

*case 1 :*

*{*

*poisk\_kol(M, N);*

*break;*

*}*

*case 2 :*

*{*

*poisk\_price(M, N);*

*break;*

*}*

*case 3 :*

*{*

*poisk\_sup(M, N);*

*break;*

*}*

*case 4 :*

*{*

*poisk\_kps(M, N);*

*break;*

*}*

*default :*

*break;*

*}*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void Poisk\_tovar(struct TOVAR \*M, int B)*

*{*

*int j;*

*while(1)*

*{*

*printf("\n");*

*for(j=0; j<6; j++)*

*printf("\n%s", poisk[j]);*

*scanf("%d",&j);*

*if (j == 0)*

*break;*

*if (j<1 || j>6)*

*continue;*

*search\_switch(M, B, j);*

*}*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

***Модуль “*** **Редактор продукции *”***

**//****edit\_product.h**

**//Интерфейс модуля *“*** **Редактор продукции *”***

*#ifndef edit\_product\_h*

*#define edit\_product\_h*

*void kor\_name(struct TOVAR \*M);*

*void kor\_kol(struct TOVAR \*M);*

*void kor\_price(struct TOVAR \*M);*

*void kor\_date(struct TOVAR \*M);*

*void kor\_sup(struct TOVAR \*M);*

*void switch\_kor\_tovar(int g, struct TOVAR \*M);*

*void Korrected\_tovar(struct TOVAR \*M, int N);*

*void poisk\_kol(struct TOVAR \*M, int B);*

*void poisk\_price(struct TOVAR \*M, int B);*

*void poisk\_sup(struct TOVAR \*M, int B);*

*void poisk\_kps(struct TOVAR \*M, int B);*

*#endif*

**//****edit\_product.cpp**

**//Реализация модуля *“*** **Редактор продукции *”***

*#include <stdio.h>*

*#include <string.h>*

*#include "base.h"*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*static char\* korrect[] =*

*{*

*" 1 > имени товара",*

*" 2 > количества товара",*

*" 3 > цены товара",*

*" 4 > срока реализации товара",*

*" 5 > поставщиков товара",*

*" 0 > назад",*

*" ? > "*

*};*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void kor\_name(struct TOVAR \*M)*

*{*

*puts(" Введите новое имя товара");*

*fflush(stdin);*

*gets(M->name);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void kor\_kol(struct TOVAR \*M)*

*{*

*puts(" Введите новое количества товара (например: 12 кг или 15 шт)");*

*scanf\_kptovar(&M->kol);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void kor\_price(struct TOVAR \*M)*

*{*

*puts(" Введите новую цену товара (например: 15 $ или 180 руб)");*

*scanf\_kptovar(&M->price);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void kor\_date(struct TOVAR \*M)*

*{*

*puts(" Введите новый срок реализации товара");*

*scanf\_date(&M->date);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void kor\_sup(struct TOVAR \*M)*

*{*

*puts(" Введите новых поставщиков товара");*

*scanf\_post\_tovar(&M->sup);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void switch\_kor\_tovar(int g, struct TOVAR \*M)*

*{*

*switch(g)*

*{*

*case 1:*

*{*

*kor\_name(M);*

*break;*

*}*

*case 2:*

*{*

*kor\_kol(M);*

*break;*

*}*

*case 3:*

*{*

*kor\_price(M);*

*break;*

*}*

*case 4:*

*{*

*kor\_date(M);*

*break;*

*}*

*case 5:*

*kor\_sup(M);*

*}*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void Korrected\_tovar(struct TOVAR \*M, int N)*

*{*

*int m, g=-1, i;*

*do*

*{*

*m=Soarch\_Element(M, N);*

*if(m >= 0)*

*{*

*while(g<0 || g>5)*

*{*

*puts("\nкорректировка:");*

*for(i=0; i<7; i++)*

*printf("\n%s", korrect[i]);*

*scanf("%d",&g);*

*if(g == 0)*

*return ;*

*printf("\n");*

*}*

*switch\_kor\_tovar(g,&M[m]);*

*puts(" Товар успешно от корректирован!");*

*}*

*m=Vyxod();*

*}*

*while(m == 1);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void poisk\_kol(struct TOVAR \*M, int B) //функция поиска товара по его количеству*

*{*

*int i, n=0;*

*struct kp\_tovar A;*

*printf(" Введите количество товара (например: 12 кг или 15 шт)\n");*

*scanf\_kptovar(&A);*

*for(i=0; i<B; i++)*

*if( sravneniya(&M[i].kol, &A) == 0)*

*n = printf\_name\_tovar(M, i);*

*if(n!=1)*

*printf(" товар не найден!\n");*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void poisk\_price(struct TOVAR \*M, int B) //функция поиска товара по его цене*

*{*

*int i, n=0;*

*struct kp\_tovar A;*

*printf(" Введите цену товара (например: 12 $ или 180 руб)\n");*

*scanf\_kptovar(&A);*

*for(i=0; i<B; i++)*

*if(sravneniya(&M[i].price, &A) == 0 )*

*n = printf\_name\_tovar(M, i);*

*if(n!=1)*

*printf(" товар не найден!\n");*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void poisk\_sup(struct TOVAR \*M, int B) //функция поиска товара по его поставщику*

*{*

*int j, i, n=0;*

*char A[80];*

*printf(" Введите поставщика товара\n");*

*fflush(stdin);*

*gets(A);*

*for(i=0; i<B; i++)*

*for(j=0; j<M[i].sup.number; j++)*

*if(0 == strcmp(A,M[i].sup.post[j]))*

*n = printf\_name\_tovar(M, i);*

*if(n!=1)*

*printf(" товар не найден!\n");*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void poisk\_kps(struct TOVAR \*M, int B) //функция поиска товара по его количеству, цене и поставщику*

*{*

*int j=0, i, l=1, h=0, m, n;*

*struct kp\_tovar a, A;*

*char S[80];*

*printf(" Введите количество товара (например: 12 кг или 15 шт)\n");*

*scanf\_kptovar(&a);*

*printf(" Введите цену товара за %s (например: 15 $ или 180 руб)\n", a.A);*

*scanf\_kptovar(&A);*

*printf(" Введите поставщика товара\n");*

*fflush(stdin);*

*gets(S);*

*for(i=0; i<B; i++)*

*{*

*n = sravneniya(&M[i].kol, &a);*

*m = sravneniya(&M[i].price, &A);*

*while(l!=0 && j<M[i].sup.number)*

*{*

*l = strcmp(S,M[i].sup.post[j]);*

*j++;*

*}*

*if( n == 0 && m == 0 && l == 0)*

*h = printf\_name\_tovar(M, i);*

*}*

*if (h == 0)*

*printf(" товар не найден!\n");*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

***Модуль “*** **База *”***

**//****base.h**

**//Интерфейс модуля *“*** **База *”***

*#ifndef base\_h*

*#define base\_h*

struct kp\_tovar //структура для количества и цены товара

{

float kp; //значени

char A[20]; //единицы измерения

};

struct post\_tovar // структура для поставщиков

{

int number; //количество поставщиков товара, max 80

char post[80][80]; // поставщики

};

struct TOVAR

{

char name[80]; //название товара

struct kp\_tovar kol; //количество товара

struct kp\_tovar price;//цена за единицу

struct tm date; //срок реализации

struct post\_tovar sup; //поставщики товара

};

*void printf\_date(struct tm \*t);*

*void printf\_kptovar(struct kp\_tovar \*f);*

*void printf\_post\_tovar(struct post\_tovar \*N);*

*int printf\_name\_tovar(struct TOVAR \*M, int i);*

*void scanf\_kptovar(struct kp\_tovar \*f);*

*void scanf\_post\_tovar(struct post\_tovar \*N);*

*int sravneniya(struct kp\_tovar \*H, struct kp\_tovar \*N);*

*int Soarch\_Element(struct TOVAR \*M, int B);*

*int prov\_date(struct tm \*t);*

*int scanf\_date(struct tm \*t);*

*int Vyxod(void);*

*#endif*

**//****base.cpp**

**//Реализация модуля *“*** **База *”***

*#include <stdio.h>*

*#include <time.h>*

*#include <string.h>*

*#include "base.h"*

*void printf\_date(struct tm \*t) //функция вывода срока реализации товара на экран*

*{*

*int tj;*

*if (9 >= t->tm\_mday)*

*printf(" 0%d.",t->tm\_mday);*

*else*

*printf(" %d.",t->tm\_mday);*

*if (9 >= t->tm\_mon)*

*printf("0%d.",t->tm\_mon);*

*else*

*printf("%d.",t->tm\_mon);*

*tj=t->tm\_year+1900;*

*printf("%d\n",tj);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void printf\_kptovar(struct kp\_tovar \*f) //функция вывода на экран цены и количества товара*

*{*

*printf(" %f", f->kp);*

*printf(" %s\n", f->A);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void printf\_post\_tovar(struct post\_tovar \*N) //функция вывода поставщиков на экран*

*{*

*int i;*

*for( i=0; i<N->number; i++)*

*printf(" %s\n", N->post[i]);*

*return ;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int printf\_name\_tovar(struct TOVAR \*M, int i)*

*{*

*printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");*

*printf("\n товар\n %s\n", M[i].name);*

*puts(" количество");*

*printf\_kptovar( &M[i].kol);*

*printf(" цена за %s\n", M[i].kol.A);*

*printf\_kptovar( &M[i].price);*

*puts(" срок реализации");*

*printf\_date(&M[i].date);*

*puts(" поставщик(и)");*

*printf\_post\_tovar(&M[i].sup);*

*return 1;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void scanf\_kptovar(struct kp\_tovar \*f) //функция ввода цены и количества товара*

*{*

*do*

*{*

*fflush(stdin);*

*scanf("%f", &f->kp);*

*scanf("%s", &f->A);*

*if(f->kp < 0)*

*printf(" значение не должно быть отрицательным!\n повторите\n");*

*}*

*while(f->kp < 0);*

*return ;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*void scanf\_post\_tovar(struct post\_tovar \*N) //функция ввода поставщиков*

*{*

*int i, n;*

*N->number = 0;*

*for (i=0; i<80; i++)*

*{*

*fflush(stdin); // очистить буфер*

*gets(N->post[i]);*

*N->number ++;*

*n=2;*

*while(n < 0 || n > 1)*

*{*

*printf("\n1 > ЕЩЕ ПОСТАВЩИК\n");*

*printf("0 > ХВАТИТ\n");*

*printf("? > ");*

*fflush(stdin);*

*scanf("%d",&n);*

*printf("\n");*

*if (n == 0)*

*return ;*

*}*

*}*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int sravneniya(struct kp\_tovar \*H, struct kp\_tovar \*N) //функция сравнения двух цен или двух количеств товара*

*{*

*if( 0 == strcmp(H->A, N->A))*

*if( N->kp == H->kp )*

*return 0; //одинаковы вернуть 0*

*return 1; //разные вернуть 1*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int Soarch\_Element(struct TOVAR \*M, int B)*

*{*

*int L = 0, R = B-1, m, flag = 0; // переменная-флаг, нашли (1) или нет (0)*

*char A[80];*

*printf("\n Введите название товара\n");*

*fflush(stdin);*

*gets( A );*

*while ( L <= R )*

*{*

*m = (L + R) / 2; // середина интервала*

*if ( 0 == strcmp(A,M[m].name ))// нашли нужный элемент*

*{*

*flag = 1; // установить флаг*

*return m; //индекс найденного элемента*

*}*

*if ( 0 > strcmp(A,M[m].name) )*

*R = m - 1; // сужаем границы области поиска*

*else*

*L = m + 1;*

*}*

*if ( flag == 0 )*

*{*

*printf ( "Такой товар не найден!\n\n" );*

*return -1;*

*}*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int prov\_date(struct tm \*t) //функция проверки правильного ввода срока реализации*

*{*

*int n=0, i;*

*if( t->tm\_mon > 12 || t->tm\_mon < 1 && t->tm\_mday < 1 || t->tm\_mday > 31 )*

*n=1;*

*switch(t->tm\_mon)*

*{*

*case 2:*

*{*

*if( t->tm\_mday > 28)*

*{*

*i = t->tm\_year%4;*

*if(i != 0 || t->tm\_mday != 29)*

*n=1;*

*}*

*break;*

*}*

*case 4:*

*case 6:*

*case 9:*

*case 11:*

*{*

*if(t->tm\_mday > 30)*

*n=1;*

*break;*

*}*

*default : break;*

*}*

*if(n)*

*{*

*printf(" Ошибка срока реализации!\n повторите\n");*

*return 1;*

*}*

*return 0;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int scanf\_date(struct tm \*t) //функция ввода срока реализации товара*

*{*

*int tj, i=1;*

*do*

*{*

*puts(" Введите число месяца (от 1 до 31)");*

*scanf("%d",&t->tm\_mday); //Число месяца(1,31)*

*puts(" Введите месяц (от 1 до 12)");*

*scanf("%d",&t->tm\_mon); // Месяцы после января(0,11)*

*puts(" Введите год");*

*scanf("%d",&tj);*

*t->tm\_year=tj-1900; //Годы с 1900(в Windows), с 1970(в Unix)*

*i=prov\_date(t);*

*}*

*while(i);*

*return 0;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*int Vyxod(void)*

*{*

*int i=2;*

*while(i < 0 || i > 1)*

*{*

*printf("\n1 > ПОВТОРИТЬ\n");*

*printf("0 > НАЗАД\n");*

*printf("? > ");*

*fflush(stdin);*

*scanf("%d",&i);*

*}*

*return i;*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*