МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА»

Институт радиоэлектроники и информационных технологий

Кафедра «Прикладная математика»

Лабораторная работа №3

Выполнила:

Студентка группы 20-ПМ-1

Бугрова Анастасия

Проверил:

Доцент кафедры «Прикладная математика»

Чернов А.Г.

Нижний Новгород

2020

**Содержание**

[**Постановка задачи** 3](#_Toc58253974)

[**Описание структуры** 4](#_Toc58253975)

[**Скриншоты работы программы** 5](#_Toc58253976)

[**Исходный код** 6](#_Toc58253977)

# **Постановка задачи**

Реализовать программу для работы с базой данных парфюмерного магазина в текстовом файле, Программа должна выполнять функции поиска, добавления, удаления и редактирования базы данных.

# **Описание структуры**

1. Структура item хранит в себе параметры конкретного товара.

Поля структуры: price – хранит в себе стоимость товара.

name- хранит название товара.

brand- хранит название товара.

2. Структура node хранит элемент списка.

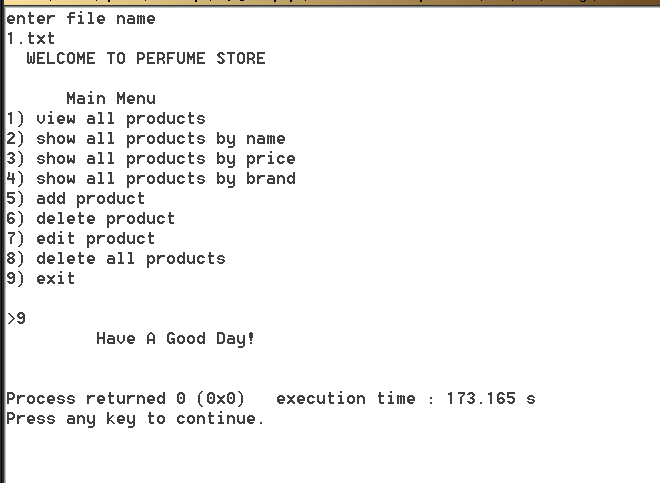
Поля структуры: структура next хранит в себе ссылку на следующий товар списка

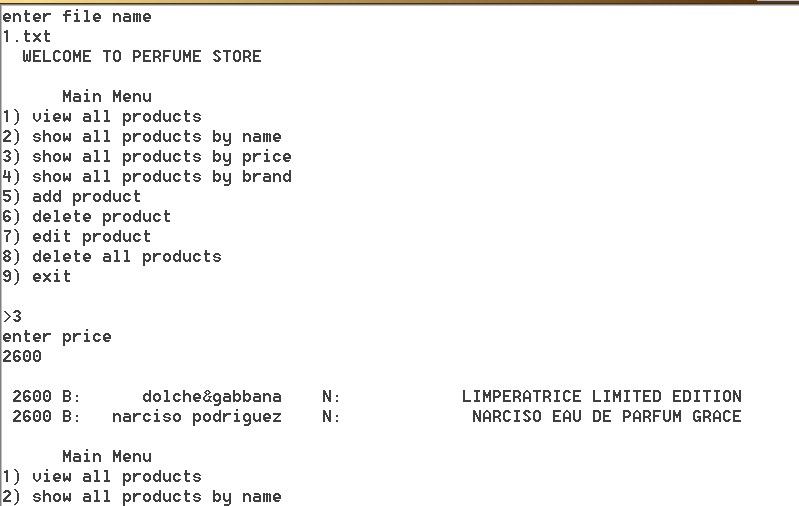
Data- хранит в себе структуру item для данного элемента.

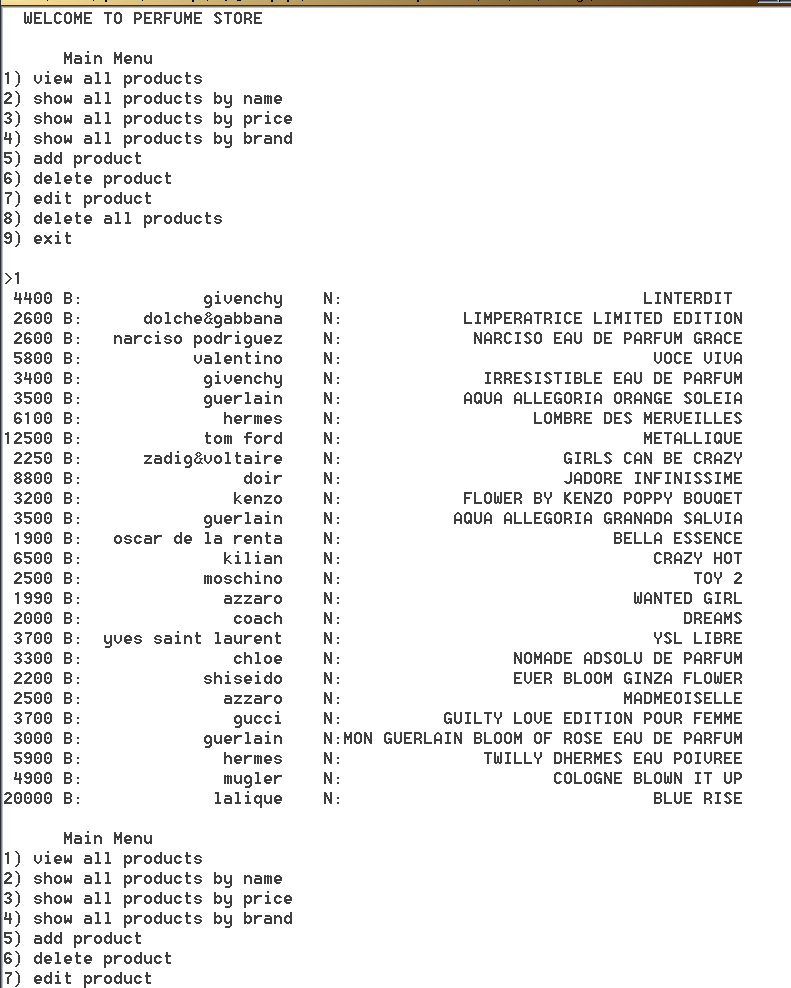
То есть данная структура хранит все данные о конкретном товаре: имя, бренд, цену и указатель на следующий товар (список товаров реализован через односвязный список).

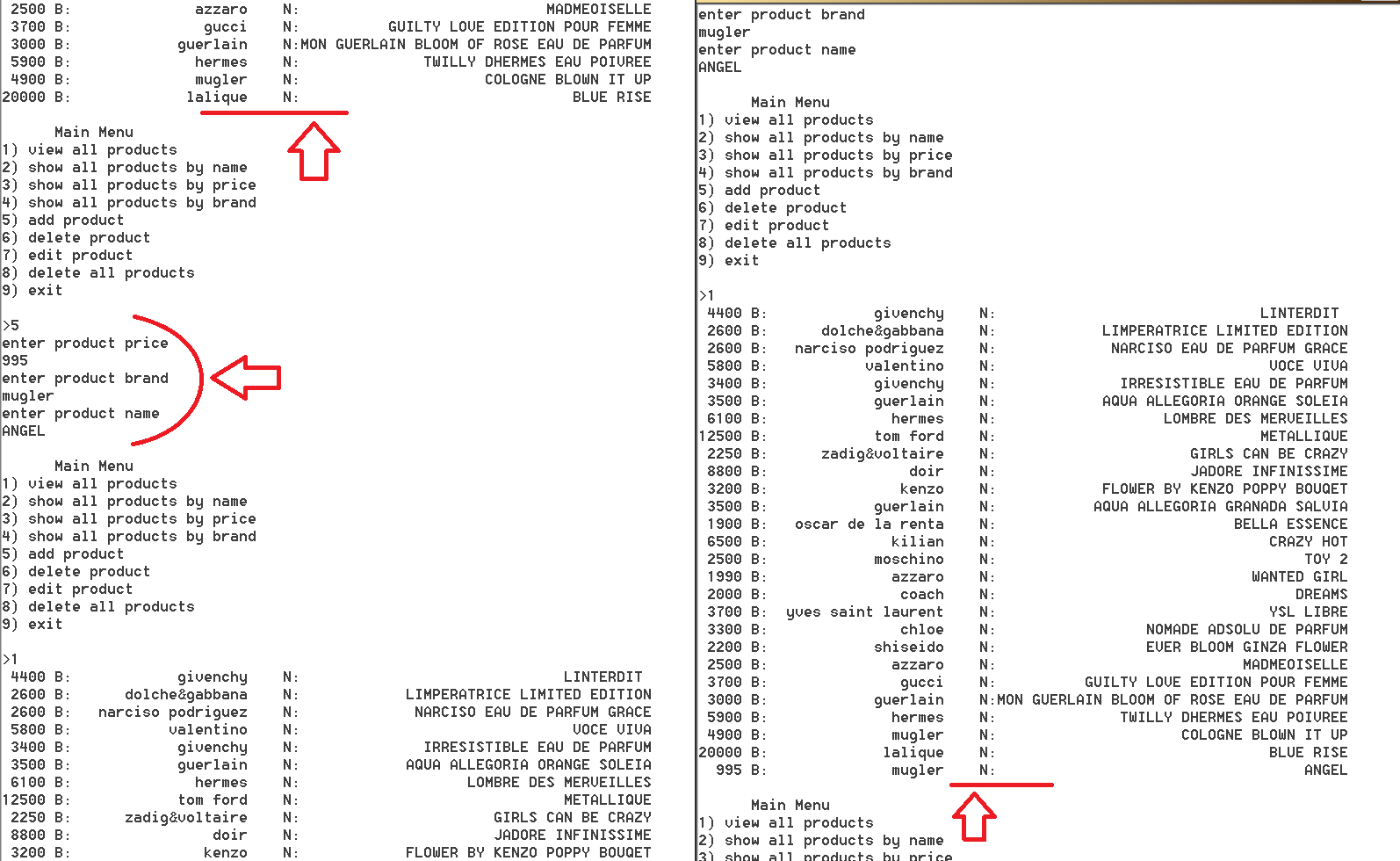
3. Структура list хранит в себе указатель на первый элемент списка товаров.

# **Скриншоты работы программы**





****

****

# **Исходный код**

Файл: function.h

#ifndef FUNCTION\_H\_INCLUDED

#define FUNCTION\_H\_INCLUDED

#include <stdlib.h>

#include "list.h"

void findByPrice(List\* Basa);

void findByBrand(List \*Basa);

void findByName(List \*Basa);

void add2basa(List\* Basa);

int deleteFromBasa(List\* Basa);

void deleteAllFromBasa(List\* Basa);

void loading(List\* L);

void change(List \*Basa);

Файл: item.h

#endif // FUNCTION\_H\_INCLUDED

#ifndef ITEM\_H\_INCLUDED

#define ITEM\_H\_INCLUDED

#include <stdio.h>

#include <stdlib.h>

struct item{

int price;

char\* name;

char\* brand;

};

typedef struct item Item;

void printt(Item i);

void freeItem(Item i);

#endif // ITEM\_H\_INCLUDED

#ifndef LIST\_H\_INCLUDED

Файл: list.h

#define LIST\_H\_INCLUDED

#include "item.h"

#include <stdio.h>

#include <stdlib.h>

char filename[100];

struct node{

struct node\* next;

Item data;

};

typedef struct node Node;

struct list {

Node\* first;

};

typedef struct list List;

void add(List\* L, Item\* foradd);

void print(List\* L);

void findPrice(List\* L,int pr);

int deleteFromList(List\* L, Item toDelete);

void write2file(List\* L);

void findBrand(List\* L,char br[40]);

void findName(List\* L,char nm[100]);

void deleteAllFromList(List\* L);

#endif // LIST\_H\_INCLUDED

Файл: util.h

#ifndef UTIL\_H\_INCLUDED

#define UTIL\_H\_INCLUDED

int stringCompare(char \*l, char \*r);

#endif // UTIL\_H\_INCLUDED

Файл: function.c

#include "function.h"

void findByPrice(List\* Basa){

printf("enter price\n");

int pr;

scanf("%d",&pr);

printf("\n");

findPrice(Basa,pr);

}

///----------------------------------------------------------------------------------

void findByBrand(List \*Basa){///

printf("enter brand\n");

char br[40];

scanf("%s",br);

printf("\n");

findBrand(Basa,br);

}

///----------------------------------------------------------------------------------

void findByName(List \*Basa){///

printf("enter name\n");

char nm[100];

scanf("%s",nm);

printf("\n");

findName(Basa,nm);

}

///----------------------------------------------------------------------------------

void loading(List \*Basa){

printf("enter file name\n");

scanf("%s", filename);

FILE\* f=fopen (filename,"r");

if (f==NULL){

printf(" !!! DATABASE CORRUPTED !!!\n");

exit(0);

}

int i,j;

Basa->first=0;

int price;

char brand[40];

char name[100];

char mm;

while(!feof(f)){

Item\* it=(Item\*)malloc(sizeof(Item));

fscanf(f,"%d",&price);

if(price<0){

free(it);

break;

}

it->price=price;

mm=fgetc(f);

i=0;

while ((mm=fgetc(f))!=':'){

brand[i]=mm;

i++;

}

brand[i]='\0';

it->brand=(char\*)malloc((i+1)\*sizeof(char));

for (j=0;j<=i;j++){

it->brand[j]=brand[j];

}

i=0;

while ((mm=fgetc(f))!='\n'){

name[i]=mm;

i++;

}

name[i]='\0';

it->name=(char\*)malloc((i+1)\*sizeof(char));

for (j=0;j<=i;j++){

it->name[j]=name[j];

}

add(Basa,it);

free(it);

}

fclose(f);

}

///---------------------------------------------------------------------------

void add2basa(List\* Basa){

int price;

char brand[40];

char name[100];

char mm;

Item\* it=(Item\*)malloc(sizeof(Item));

int i,j;

printf("enter product price \n");

scanf("%d",&price);

it->price=price;

getchar();

printf("enter product brand\n");

i=0;

while ((mm=getchar())!='\n'){

brand[i]=mm;

i++;

}

brand[i]='\0';

it->brand=(char\*)malloc((i+1)\*sizeof(char));

for (j=0;j<=i;j++){

it->brand[j]=brand[j];

}

printf("enter product name\n");

i=0;

while ((mm=getchar())!='\n'){

name[i]=mm;

i++;

}

name[i]='\0';

it->name=(char\*)malloc((i+1)\*sizeof(char));

for (j=0;j<=i;j++){

it->name[j]=name[j];

}

add(Basa,it);

free(it);

}

///---------------------------------------------------------------------------------------

int deleteFromBasa(List\* Basa){

int price;

char brand[40];

char name[100];

char mm;

Item\* it=(Item\*)malloc(sizeof(Item));

int i,j;

printf("enter product price \n");

scanf("%d",&price);

it->price=price;

getchar();

printf("enter product brand\n");

i=0;

while ((mm=getchar())!='\n'){

brand[i]=mm;

i++;

}

brand[i]='\0';

it->brand=(char\*)malloc((i+1)\*sizeof(char));

for (j=0;j<=i;j++){

it->brand[j]=brand[j];

}

printf("enter product name\n");

i=0;

while ((mm=getchar())!='\n'){

name[i]=mm;

i++;

}

name[i]='\0';

it->name=(char\*)malloc((i+1)\*sizeof(char));

for (j=0;j<=i;j++){

it->name[j]=name[j];

}

return deleteFromList(Basa,\*it);

}

///---------------------------------------------------------------------

void deleteAllFromBasa(List\* Basa){

deleteAllFromList(Basa);

}

///-----------------------------------------------------------------------

void change(List\* Basa){

int result=deleteFromBasa(Basa);

if (result==1){

printf("now enter new product options\n");

add2basa(Basa);

}

}

Файл: item.c

#include "item.h"

void printt(Item i){

printf("%5d B:%20s\tN:%40s\n",i.price,i.brand,i.name);

}

void freeItem(Item i){

free(i.brand);

free(i.name);

}

Файл: list.c

#include "util.h"

#include "list.h"

///----------------------------------------------------------------------------------

void add(List \*L,Item \*foradd){

if (L->first==0){

L->first=(Node\*)malloc(sizeof (Node));

L->first->data = \*foradd;

L->first->next=0;

}

else{

Node \*tmp=L->first;

while(tmp->next!=0){

tmp=tmp->next;

}

tmp->next=(Node\*)malloc(sizeof (Node));

tmp->next->data = \*foradd;

tmp->next->next=0;

}

}

///----------------------------------------------------------------------------------

void print(List \*L){

Node \*tmp=L->first;

while (tmp!=0){

printt(tmp->data);

tmp=tmp->next;

}

}

///----------------------------------------------------------------------------------

void findPrice(List\* L,int pr){

Node \*tmp=L->first;

int k=0;

while (tmp!=0){

if (tmp->data.price==pr){

printt(tmp->data);

k=1;

}

tmp=tmp->next;

}

if (k==0){

printf("THERE IS NO SUCH PRODUCTS\n");

}

}

///----------------------------------------------------------------------------------

int deleteFromList(List\* L, Item toDelete){

Node \*tmp=L->first;

//printf("---");

//printt(tmp->data);

//printt(toDelete);

//printf("%i %i\n", stringCompare(tmp->data.brand, toDelete.brand ), stringCompare(tmp->data.name, toDelete.name));

if (tmp->data.price==toDelete.price &&

stringCompare(tmp->data.brand, toDelete.brand )==0 &&

stringCompare(tmp->data.name, toDelete.name )==0 )

{

//printf("==============");

Node \*remove = L->first;

// новый первый элемент теперь второй

L->first = L->first->next;

// освобождаем память

freeItem(remove->data);

free(remove);

return 1;

}

Node \*prev=tmp;

tmp = prev->next;

while (tmp!=0){

//printf("---\n");

//printt(tmp->data);

//printt(toDelete);

if (tmp->data.price==toDelete.price &&

stringCompare(tmp->data.brand, toDelete.brand )==0 &&

stringCompare(tmp->data.name, toDelete.name )==0 )

{

Node \*remove = tmp;

// новый первый элемент теперь второй

prev->next = tmp->next;

// освобождаем память

freeItem(remove->data);

free(remove);

return 1;

}

prev = tmp;

tmp =prev->next;

}

printf("THERE IS NO SUCH PRODUCT\n");

return 0;

}

///----------------------------------------------------------------------------------

void write2file(List\* L){

FILE\* f=fopen(filename,"w");

Node \*tmp=L->first;

while (tmp!=0){

fprintf (f,"%d:%s:%s\n",tmp->data.price,tmp->data.brand,tmp->data.name);

tmp=tmp->next;

}

fprintf(f,"-1");

fclose(f);

}

///----------------------------------------------------------------------------------

void deleteAllFromList(List\* L){

Node\* tmp = L->first;

while(L->first != 0){

tmp = L->first->next;

freeItem(L->first->data);

//printf("%d %d \n", &L->first->data , L->first);

free(L->first);

L->first=tmp;

}

return;

}

///----------------------------------------------------------------------------------

void findBrand(List\* L,char br[40]){///

Node \*tmp=L->first;

int k=0;

while (tmp!=0){

if (stringCompare(tmp->data.brand,br)==0){

printt(tmp->data);

k=1;

}

tmp=tmp->next;

}

if (k==0){

printf("THERE IS NO SUCH PRODUCTS\n");

}

}

///----------------------------------------------------------------------------------

void findName(List\* L,char nm[100]){///

Node \*tmp=L->first;

int k=0;

while (tmp!=0){

if (stringCompare(tmp->data.name,nm)==0){

printt(tmp->data);

k=1;

}

tmp=tmp->next;

}

if (k==0){

printf("THERE IS NO SUCH PRODUCTS\n");

}

}

///----------------------------------------------------------------------------------

Файл: main.c

#include <stdio.h>

#include <stdlib.h>

#include "list.h"

#include "function.h"

// listOfGoods

int main()

{

/// "listOfGoods.txt"

List\* Basa=(List\*)malloc(sizeof(List));

loading(Basa);

printf (" WELCOME TO PERFUME STORE\n");

getchar();

char num=0;char musor;

while (num!='9'){

printf("\n");

printf(" Main Menu\n");

printf("1) view all products \n");

printf("2) show all products by name \n");

printf("3) show all products by price \n");

printf("4) show all products by brand \n");

printf("5) add product \n");

printf("6) delete product \n");

printf("7) edit product \n");

printf("8) delete all products \n");

printf("9) exit \n");

printf("\n");

printf(">");

while((num=getchar())=='\n'){///

}

musor=getchar();

//printf("a=%d",a);

switch (num){

case '1':

print(Basa);

break;

case '2':

findByName(Basa);

break;

case '3':

findByPrice(Basa); //getchar();

break;

case '4':

findByBrand(Basa);

break;

case '5':

add2basa(Basa);

break;

case '6':

deleteFromBasa(Basa);

break;

case '7':

change(Basa);;

break;

case '8':

deleteAllFromBasa(Basa);

break;

case '9':

break;

default:

printf("please choose correct number");

}

}

write2file(Basa);

free(Basa);

printf("\t Have A Good Day!\n\n");

return 0;

}

Файл: util.c

#include "util.h"

#include <string.h>

int stringCompare(char \*l, char \*r){

//двигаем указатели пока символы равны

while(\*(l++) == \*(r++)){if(\*l=='\0' || \*r=='\0') break;}

// отдаем разность кодов символов, 0 - равны, -1 левая меньше, 1 правая

// меньше

return \*l - \*r;

//return strcmp(l,r);

}