

Министерство образования Российской Федерации  
Пензенский государственный университет  
Кафедра «Вычислительная техника»

## **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе  
по курсу «Программирование»  
на тему «Разработка программы сложной структуры методом  
нисходящего программирования»

Выполнил: Митрошин М.А.

Студент группы 21ВО1

Принял: Митрохин М.А.

Зав. Кафедрой ВТ

Пенза, 2022



## Содержание

|                                               |    |
|-----------------------------------------------|----|
| Введение .....                                | 3  |
| 1. Постановка задачи .....                    | 3  |
| 2. Выбор решения.....                         | 4  |
| 3. Описание разработки программы.....         | 5  |
| 4. Отладка и тестирование .....               | 6  |
| 5. Описание программы .....                   | 6  |
| 5.1 Разработка меню.....                      | 7  |
| 6. Руководство пользователя .....             | 16 |
| Заключение .....                              | 18 |
| Список используемых источников.....           | 18 |
| Приложение А. Листинг программы.....          | 19 |
| Приложение В. Результат работы программы..... | 28 |
| Приложение С. Результат работы программы..... | 30 |

## Введение

В процессе накопления знаний увеличивается количество информации, представляющей собой меру измерения знаний. Для познания человека требуется обработать всю накопленную за прошедшее время информацию, вы брать из неё самое важное. Существующие сейчас методы получения информации -

это, в основном, компьютеры и вычислительные устройства, хранящие, обрабатывающие и передающие информацию.

Однако же программа по-прежнему зависит от человека, и то, как он её напишет и для чего, определит дальнейшие функции и успешное применение.

Компьютер стал заменой для многих вещей, и внедрение его в различные сферы жизни общества помогает ему встать на место тех вещей, которые он может заменить.

Задачей данной курсовой работы стала визуализация базы данных «Накладная». Разработанная программа, предназначенная для работы с базой данных

«Накладная», может использоваться людьми, не имеющими квалификации программиста во многих сферах, например, торговой. Программа разработана с использованием современных концепций программирования - структурного программирования языка Си. [1,2,3]

Пояснительная записка содержит техническое задание, описание программы, программу и методику испытаний, описание применения, текст программы.

### 1. Постановка задачи

Необходимо разработать программу для работы с базой данных «Накладная». Программа содержит следующие поля: наименование товара, единица измерения, цена, количество товара, сумма.

Программа должна осуществлять создание базы данных, вывод ее для просмотра пользователем, а также добавление, удаление, поиск, сортировку, чтение из файла, запись в файл и удаление файла.

Необходимо разработать алгоритм работы меню консоли и продумать структуру программы и взаимодействие всех входящих в нее функций.

**Многомодульность программы.** Программа должна быть поделена на логические модули, что позволяет быстрее проводить отладку и тестирование, находить возможные ошибки и расширять функционал.

**Использование сложных типов данных – списков, структур, файлов.** Использование односвязного списка и структур необходимо для упрощения работы с данными в коде. Использование файлов необходимо для сохранения базы данных и дальнейшего её открытия.

**Режим работы программы – текстовый.** Использование программы происходит в текстовом формате посредством консоли.

**Интерфейс** должен быть построен на основе меню. Необходимо изучить приемы разработки меню для создания интуитивно понятного пользовательского интерфейса.

**Наличие заставки.** Пользователю, запустившему программу, необходима заставка, дающая сведения о базе данных и ее возможности перехода к работе.

## 2. Выбор решения

При запуске программы открывается приветственный экран. Далее появляется меню, состоящее из восьми пунктов:

- 1) Создание базы данных. Пользователь заполняет поля: «Наименование», «Единица измерения», «Цена», «Количество», «Сумма». Если база данных была создана ранее, то программа перезапишет данные.
- 2) Добавление элементов. Пользователь вводит количество записей, которое хочет добавить. Далее заполняет поля (см. Создание базы данных). Новые записи добавляются в конец к уже созданной базе данных. Если база данных пуста, выводится сообщение об ошибке.
- 3) Просмотр базы данных. Если база данных существует, то она выводится на экран. Иначе сообщение об ошибке.
- 4) Удаление элементов. Пользователь вводит номер записи, которую желает удалить. Выводится сообщение об успешном удалении. Если записи не существует или база данных не заполнена, выводится сообщение об ошибке.
- 5) Удаление элементов. Пользователь вводит номер записи, которую желает удалить. Выводится сообщение об успешном удалении. Если записи не существует или база данных не заполнена, выводится сообщение об ошибке.
- 6) Поиск элементов. Пользователь вводит номер записи, которую хочет найти. Если запись существует, она выведется на экран. Если записи не существует или база данных не заполнена, выводится сообщение об ошибке.

7) Открытие и сохранение файлов. Пользователь выбирает действие, которое собирается произвести.

7.1 Открытие. Если файл, являющийся аргументом командной строки существует, выводится соответствующее сообщение, а с базой данных можно производить все вышеперечисленные действия. Если файл не существует, выводится сообщение об ошибке.

7.2 Сохранение. Если файл, являющийся аргументом командной строки существует, данные в нем перезаписываются. Если база данных пуста, то сохранять нечего, соответственно, выводится сообщение об ошибке.

7.3 Удаление. При выборе данного пункта происходит удаление файла.

8) Выход. Выход из программы.

### 3. Описание разработки программы

Разработанная программа состоит из нескольких модулей:

- 1) main.c — включает вывод заставки и само меню.
- 2) Add.c — включает функцию добавления элемента в базу.
- 3) Delete.c — включает функцию удаления элемента из базы данных.
- 4) Print.c — включает функцию вывода базы данных на экран.
- 5) SaveToFile.c — включает функцию сохранения базы данных в файл.
- 6) LoadFromFile.c — включает функцию загрузки базы данных из файла.
- 7) Remove.c — включает функцию удаления файла.
- 8) Search.c — включает функцию поиска элемента в базе данных.

### 4. Отладка и тестирование

В качестве среды разработки была выбрана программа Visual Studio 2022.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы

### 5. Описание программы

Приложение main.exe является основным модулем программы. При запуске программы происходит вывод начальной заставки, после чего пользователю необходимо выбрать пункт меню для дальнейших действий над базой данных. Выход из меню осуществляется после выбора пользователем пункта «Завершить выполнение программы - 0». Описание состояний программы выполнено в таблице ниже.

| Действие пользователя | Действия программы    |
|-----------------------|-----------------------|
| Выбран пункт          | Программа запрашивает |

|                                               |                                                                      |
|-----------------------------------------------|----------------------------------------------------------------------|
| «добавить товар»                              | информацию о товаре и добавляет его в базу данных                    |
| Выбран пункт «удалить товар»                  | Программа запрашивает номер элемента и удаляет его                   |
| Выбран пункт «вывод товаров»                  | Программа выводит всю базу данных на экран                           |
| Выбран пункт «сохранить файл»                 | Программа создаёт файл и загружает в неё базу данных                 |
| Выбран пункт «загрузить из файла»             | Программа загружает в базу данных информацию из файла                |
| Выбран пункт «поиск товара»                   | Программа запрашивает наименование товара и выводит найденные записи |
| Выбран пункт «Завершить работу программы - 0» | Закрытие программы                                                   |

**Таблица 1**

## **5.1 Разработка меню**

Меню реализовано в `main.cpp` и состоит из следующих пунктов: «добавить товар», «удалить товар», «вывод товаров», «поиск по имени», «сохранить в файл», «загрузить из файла», «удалить файл». Главной задачей функций, реализованных в файле, является предоставление пользователю возможности выбрать один из пунктов меню и осуществить необходимые действия в программе.

На рис. 1 приведена функция `main`

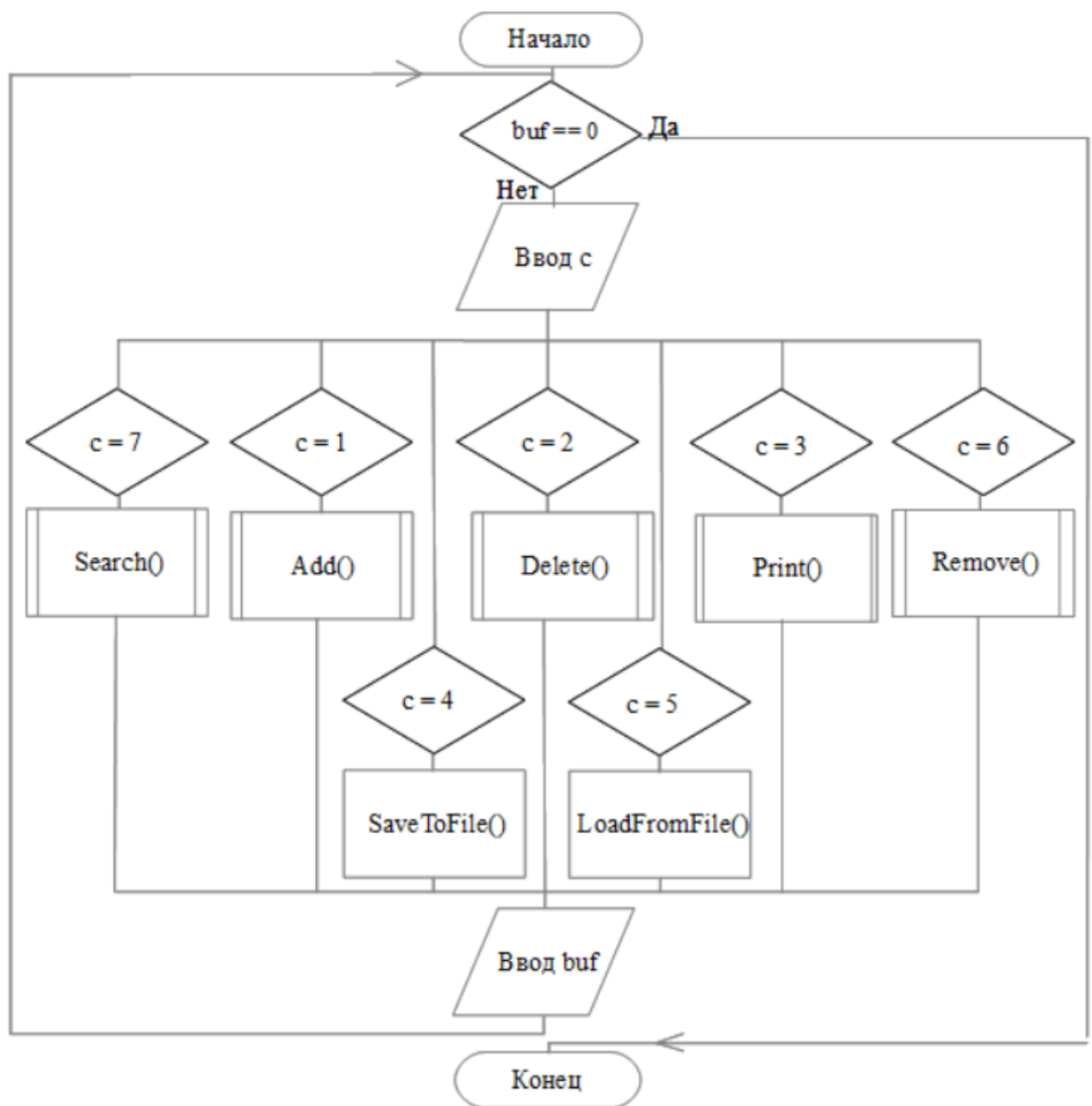


Рисунок 1 – Блок-схема алгоритма

На рис. 2 приведена функция add

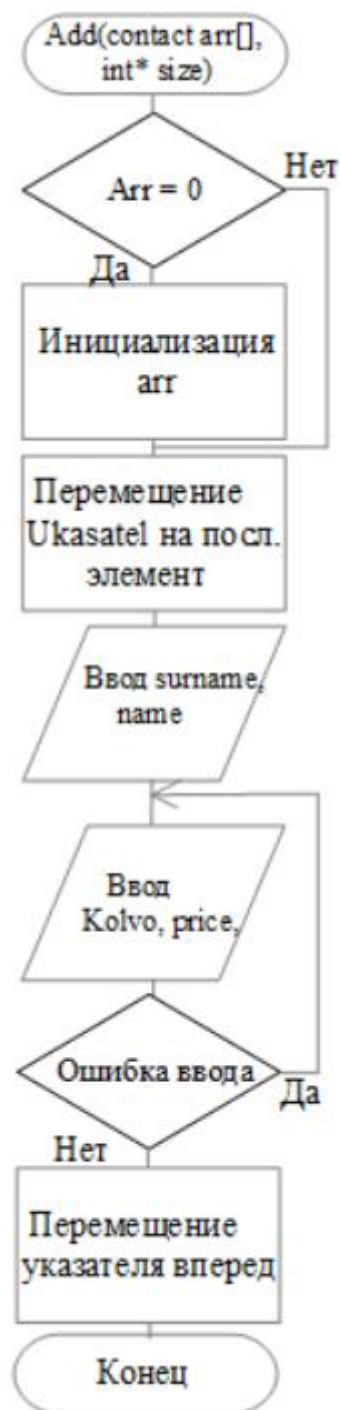


Рисунок 2 - Блок-схема алгоритма

На рис. 3 приведена функция delete

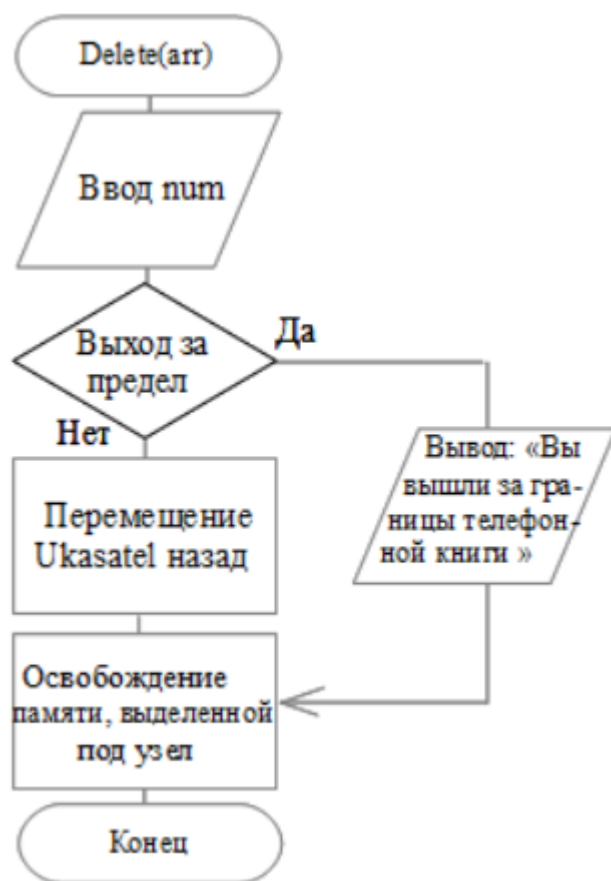


Рисунок 3- Блок-схема алгоритма

На рис. 4 приведена функция print

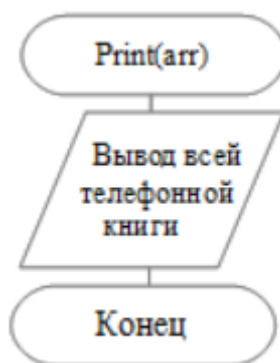


Рисунок 4 - Блок-схема алгоритма

На рис. 5 приведена функция savetofile



**Рисунок 5- Блок-схема алгоритма**

На рис. 6 приведена функция `loadfromfile`

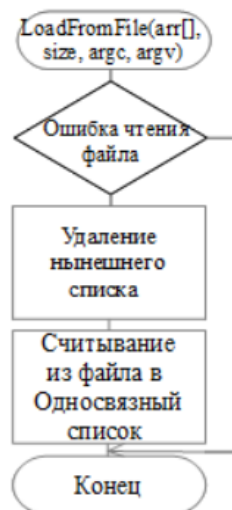


**Рисунок 6 - Блок-схема алгоритма**

На рис. 7 приведена функция `remove`

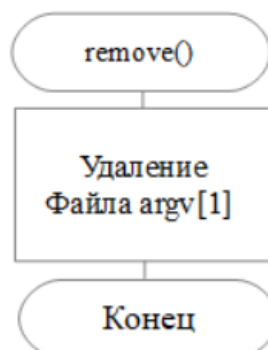


**Рисунок 7 - Блок-схема алгоритма**

На рис. 8 приведена функция `search`

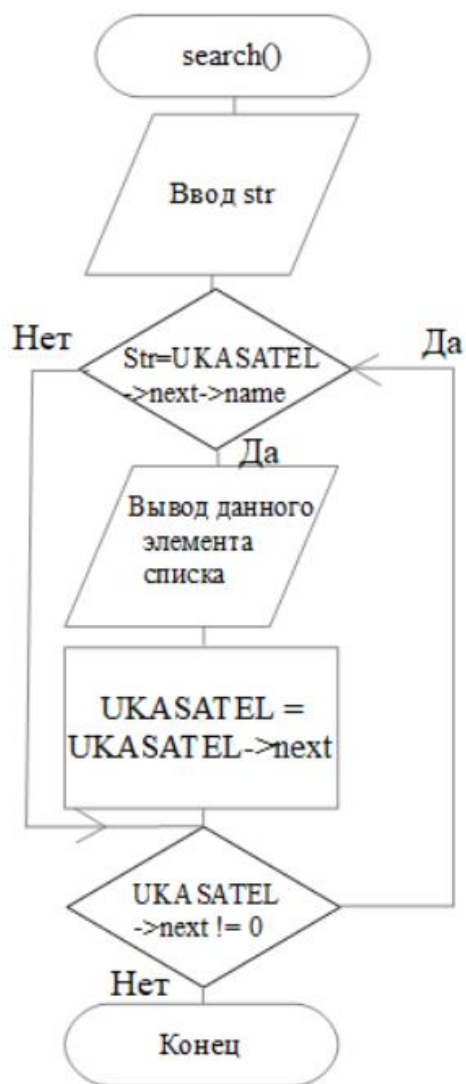


Рисунок 8 - Блок-схема алгоритма

## 6. Руководство пользователя

Программа main.exe предназначена для хранения информации о поступлении товаров на склад (наименование, единица измерения, цена, количество, сумма). Программа имеет интуитивно понятный интерфейс и может осуществлять такие операции, как создание и просмотр базы данных, добавление, поиск, удаление, а также сохранение базы данных в файл и чтение её из файла.

При запуске программы на экран выводится главная заставка и через пять секунд программа переходит в главное меню.

-----  
Курсовая работа - Накладная  
Сделал - Митрошин М.А.  
Студент группы 21В01  
Принял - Митрохин М.А.  
Зав. Кафедрой ВТ  
-----

- ```
Введите наименование товара: moloکو
Введите единицу измерения товара: litr
Введите цену товара: 55
Введите количество товара: 5

Продолжить работу программы? (0 - Да, 1 - Нет)0
```

Затем нужно ввести 0 или 1, в зависимости от того, нужно ли продолжать работу программы. Введём ещё данных и выведем их, введя цифру 3.

```
1 - добавить товар
2 - удалить товар
3 - вывод товаров
4 - сохранить в файл
5 - загрузить из файла
6 - удалить файл
7 - поиск по имени
3

№1      moloko  liter  55      5      275
№2      hleb    shuka  30      70     2100
№3      banan   kg     50      20     1000
№4      voda    liter  25      20     500
Продолжить работу программы? (0 - Да, 1 - Нет)
```

Для того, чтобы удалить запись, необходимо выбрать соответствующий пункт меню. Введём номер товара и выведем результат.

```
7 - поиск по имени
2

Введите номер товара, который вы хотите удалить: 1
Продолжить работу программы? (0 - Да, 1 - Нет)0

1 - добавить товар
2 - удалить товар
3 - вывод товаров
4 - сохранить в файл
5 - загрузить из файла
6 - удалить файл
7 - поиск по имени
3

№1      hleb    shuka  30      70     2100
№2      banan   kg     50      20     1000
№3      voda    liter  25      20     500
Продолжить работу программы? (0 - Да, 1 - Нет)
```

Далее выберем пункт “поиск по имени” и найдём наименование одного из товаров

```
1 - добавить товар
2 - удалить товар
3 - вывод товаров
4 - сохранить в файл
5 - загрузить из файла
6 - удалить файл
7 - поиск по имени
7

Введите наименование товара: banan
Найденные записи
№2      banan   kg      50      20      1000
```

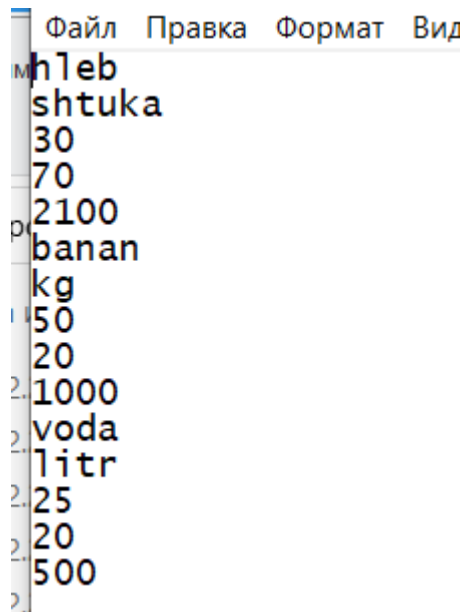
Сохраним всю базу данных, выбрав соответствующий пункт в меню

```
1 - добавить товар
2 - удалить товар
3 - вывод товаров
4 - сохранить в файл
5 - загрузить из файла
6 - удалить файл
7 - поиск по имени
4

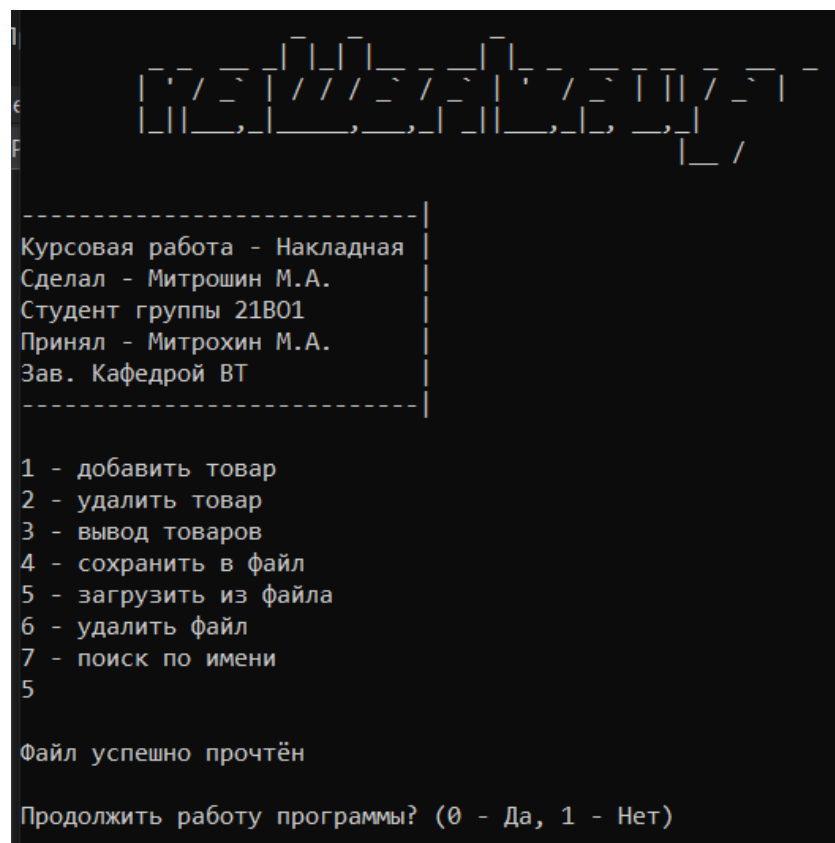
Успешно записано в файл

Продолжить работу программы? (0 - Да, 1 - Нет)
```

Имя файла является аргументом командной строки. Сам файл хранится в том же месте, где и программа. Проверим этот файл.



Теперь, запустив программу заново, загрузим данные из файла в базу данных, выбрав пятый пункт меню.



Выведем загруженную базу данных на экран.



| Имя                      | Состояние | Дата изменения   | Тип                    | Размер |
|--------------------------|-----------|------------------|------------------------|--------|
| .vs                      | ✗         | 24.12.2022 21:25 | Папка с файлами        |        |
| x64                      | ✗         | 24.12.2022 21:25 | Папка с файлами        |        |
| add.c                    | ✗         | 27.12.2022 18:19 | C Source               | 4 КБ   |
| add.h                    | ✗         | 25.12.2022 13:04 | C/C++ Header           | 1 КБ   |
| delete.c                 | ✗         | 27.12.2022 21:54 | C Source               | 2 КБ   |
| delete.h                 | ✗         | 24.12.2022 22:06 | C/C++ Header           | 1 КБ   |
| loadfromfile.c           | ✗         | 27.12.2022 18:38 | C Source               | 5 КБ   |
| loadfromfile.h           | ✗         | 24.12.2022 22:11 | C/C++ Header           | 1 КБ   |
| main.c                   | ✗         | 27.12.2022 22:05 | C Source               | 4 КБ   |
| print.c                  | ✗         | 25.12.2022 18:06 | C Source               | 1 КБ   |
| print.h                  | ✗         | 24.12.2022 22:06 | C/C++ Header           | 1 КБ   |
| Project3.sln             | ✗         | 23.12.2022 18:53 | Visual Studio Solut... | 2 КБ   |
| Project3.vcxproj         | ✗         | 26.12.2022 18:45 | VC++ Project           | 8 КБ   |
| Project3.vcxproj.filters | ✗         | 26.12.2022 18:45 | VC++ Project Filte...  | 3 КБ   |
| Project3.vcxproj.user    | ✗         | 26.12.2022 13:20 | Per-User Project O...  | 1 КБ   |
| remove.c                 | ✗         | 27.12.2022 22:31 | C Source               | 1 КБ   |
| remove.h                 | ✗         | 26.12.2022 12:43 | C/C++ Header           | 1 КБ   |
| savetofile.c             | ✗         | 27.12.2022 18:33 | C Source               | 1 КБ   |
| savetofile.h             | ✗         | 24.12.2022 22:06 | C/C++ Header           | 1 КБ   |
| search.c                 | ✗         | 27.12.2022 22:14 | C Source               | 2 КБ   |
| search.h                 | ✗         | 26.12.2022 12:43 | C/C++ Header           | 1 КБ   |

Файл действительно удалён.

## Заключение

В ходе выполнения данной курсовой работы были получены навыки разработки многомодульных программ. Была освоена работа с файлами, изучены функции работы с консолью и оперативной памятью, освоены приемы создания пользовательского интерфейса. Были изучены темы: работа с памятью, работа с файлами, многомодульное программирование. Также были получены основные навыки работы в среде Microsoft Visual Studio 2019 на языке Си.

В рамках курсовой работы была написана программа для работы с базой данных «Накладная». Она имеет все основные функции для базы данных.

## Список используемых источников

1. Б.В. Керниган, Д.М. Ритчи «Язык программирования Си» 1985 г.
2. Программирование на С и С++, Онлайн справочник программиста на С и С++.
3. METANIT.COM, Сайт о программировании
4. Документация по Microsoft C++, С и ассемблеру

## Приложение А. Листинг программы

### Приложение А.1 Файл «main.c»

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#include"add.h"
#include"time.h"
#include"delete.h"
#include"loadfromfile.h"

void main(int argc, char* argv[]) {
int msec = 0, trigger = 5000; /* 10ms */
clock_t before = clock();
int iterations = 0;
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    printf("
printf("
printf("
printf("
printf("
printf("-----|\n");
printf("Курсовая работа - Накладная |\n");
printf("Сделал - Митрошин М.А. |\n");
printf("Студент группы 21В01 |\n");
printf("Принял - Митрохин М.А. |\n");
printf("Зав. Кафедрой ВТ |\n");
printf("-----|\n");
```

```

do {
clock_t difference = clock() - before;
    msec = difference * 1000 / CLOCKS_PER_SEC;
    iterations++;
} while (msec < trigger && !_kbhit());

SetConsoleCP(1251);
SetConsoleOutputCP(1251);

int size = 0, buf = 0, durak = 0;
char c;
structnaklad* nakl = NULL;
if (argc == 1) {
    printf("не обнаружено имя файла\n");
    exit(1);
}

while (buf == 0) {
    durak = 0;
while (durak == 0) {
durak = 0;
        printf("\n1 - добавить товар\n2 - удалить товар\n3 - вывод товаров\n4
- сохранить в файл\n5 - загрузить из файла\n6 - удалить файл\n7 - поиск по
имени\n");
scanf("%c", &c);
if ((c >= '1') && (c <= '8'))
    durak = 1;
else {
        durak = 0;
        printf("\nОШИБКА ВВОДА\n");
        getchar();
    }
    }

    printf("\n");
switch (c) {
case'1': nakl = Add(nakl);
break;
case'2': nakl = Delete(nakl);
break;
case'3': Print(nakl);
break;
case'4': SaveToFile(nakl, argc, argv);
break;
case'5': nakl = LoadFromFile(nakl, argc, argv);
break;
case'6': Remove(argc, argv);
break;
case'7': Search(nakl);

        }
        printf("Продолжить работу программы? (0 - Да, 1 - Нет)");
scanf("%d", &buf); getchar();
printf("\n");
    }
if (nakl != NULL)
{
structnaklad* UKASATEL = nakl;
do
    {
        UKASATEL = nakl->next;
        free(nakl->edizm);
        free(nakl->name);
//    free(nakl->price);
    }
}

```

```

//    free(nakl->sum);
        free(nakl);
        nakl = UKASATEL;
    } while (nakl != NULL);
    free(UKASATEL);
}
free(nakl);
system("pause");
}

```

## Приложение А.2 Файл «delete.c»

```

#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include"add.h"
#include"delete.h"

structnaklad* Delete(structnaklad* arr) {
structnaklad* UKASATEL = arr;
int num;
    UKASATEL = arr;
if (arr == NULL)
{
    printf("Вы вышли за границы\n");
returnNULL;
}
if ((UKASATEL != NULL))
{
    printf("Введите номер товара, который вы хотите удалить: ");
scanf("%d", &num);
if (num < 1)
{
    printf("Вы вышли за границы\n");
returnarr;
}
if (num == 1)
{
structnaklad* del = arr;
arr = UKASATEL->next;
    free(del->edizm);
    free(del->name);
    free(del->price);
    free(del);
returnarr;
}
for (int i = 1; i < num-1 ; i++)
{
    UKASATEL = UKASATEL->next;
if (UKASATEL->next == NULL&& i != num - 1)
{
    printf("Вы вышли за границы\n");
returnarr;
}
}
structnaklad* del = UKASATEL->next;
    UKASATEL->next = UKASATEL->next->next;
    UKASATEL->next = del->next;
    free(del->edizm);
    free(del->name);a

```

```

        free(del->price);
        free(del);
return arr;
    }
else printf("Вышли за границы\n");

}

```

## Приложение А.3 Файл «print.c»

```

#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include"add.h"
#include"print.h"
void Print(struct naklad* arr) {
    struct naklad* UKASATEL = arr;
    int i = 1;
    if (UKASATEL == NULL)
    {
        printf("\n%s\n", "СТРУКТУРА ПУСТА");
    }
    return;
}

do
{
    printf("№%d\t%s\t%s\t%s\t%s\t%s\t\n", i, UKASATEL->name, UKASATEL->edizm,
        UKASATEL->price, UKASATEL->kolvo, UKASATEL->sum);
    i++;
    UKASATEL = UKASATEL->next;
} while (UKASATEL != NULL);
}

```

## Приложение А.4 Файл «savetofile.c»

```

#define log_info(M, ...) fprintf(\ stderr, "[INFO] (%s:%d) " M "\n", \ __FILE__,
    __LINE__, ## __VA_ARGS__)
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include"add.h"
#include"savetofile.h"

int SaveToFile(struct naklad* arr, int argc, char** argv) {
    FILE* outfile;
    struct naklad* UKASATEL = arr;
    if (UKASATEL != NULL)
    {
        if ((outfile = fopen(argv[1], "wt")) == NULL)
        {
            fprintf(stderr, "ОШИБКА ПРИ ОТКРЫТИИ ФАЙЛА\n");
            return 1;
        }
        do
        {
            fprintf(outfile, "%s\n%s\n%s\n%s\n%s\n", UKASATEL->name, UKASATEL->edizm,
                UKASATEL->price, UKASATEL->kolvo, UKASATEL->sum);
            UKASATEL = UKASATEL->next;
        } while (UKASATEL != NULL);
        fclose(outfile);
        printf("Успешно записан файл\n\n");
    }
    else fprintf(stderr, "СТРУКТУРА ПУСТА\n");
}

```

```
return 0;

}
```

## Приложение А.5 Файл «remove.c»

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include"add.h"
#include"print.h"
#include"remove.h"
void Remove(int argc, char** argv) {
    remove("tet.txt");
    /*
    remove(argv[1]);
    if (-1 == remove(argv[1]))
        printf("Ошибка\n");
    else*/
    printf("Файл удалён\n");
}
```

## Приложение А.6 Файл «loadfromfile.c»

```
#define log_info(M, ...) fprintf(\ stderr, "[INFO] (%s:%d) " M "\n", \ __FILE__,
__LINE__, ## VA_ARGS)
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include"add.h"
#include"loadfromfile.h"

struct naklad* LoadFromFile(struct naklad* arr, int argc, char** argv)
{
    int j = 0;
    struct naklad* UKASATEL = arr;
    if (arr != NULL)
    {
        do
        {
            UKASATEL = arr->next;
            free(arr->edizm);
            free(arr->name);
            free(arr->price);
            free(arr);
        } while (arr != NULL);
        arr = UKASATEL;
    }
    int i = 1;
    FILE* f = NULL;
    f = fopen(argv[1], "rt");
    if (!f) {
        fprintf(stderr, "ОШИБКА ПРИ ОТКРЫТИИ ФАЙЛА");
        exit(1);
    }
    if (f == NULL)
    {
        printf("Error!!! File not found\n\n");
        getchar();
    }
    int sizest = 0;
    char buf;
    arr = NULL;
    if (arr == NULL)
```

```

    {
arr = malloc(sizeof(structnaklad));
arr->next = NULL;
        UKASATEL = arr;
arr = NULL;
    }
while (!feof(f))
    {
if (!feof(f))
    {
while (1)
        {
if (UKASATEL->next != NULL)
            UKASATEL = UKASATEL->next;
elsebreak;
        }

int sizest = 0;
char buf;
            UKASATEL->next = malloc(sizeof(structnaklad));
            UKASATEL->next->name = calloc(1, sizeof(char));
while (1)
        {
            buf = fgetc(f);
if (buf == -1)
            {
                free(UKASATEL->next->name);
                free(UKASATEL->next);
                UKASATEL->next = NULL;
returnarr;
            }
            UKASATEL->next->name = realloc(UKASATEL->next->name, (sizest + 1)
* sizeof(char));
if ((UKASATEL->next->name[sizest] = buf) == '\n')
break;
                sizest++;
            }
            UKASATEL->next->name[sizest] = '\0';
            UKASATEL->next->edizm = calloc(1, sizeof(char));
//////////
            sizest = 0;
while (1)
        {
            buf = fgetc(f);
            UKASATEL->next->edizm = realloc(UKASATEL->next->edizm, (sizest +
1) * sizeof(char));
if ((UKASATEL->next->edizm[sizest] = buf) == '\n')
break;
                sizest++;
            }
            UKASATEL->next->edizm[sizest] = '\0';
//////////
            sizest = 0;
            UKASATEL->next->price = calloc(1, sizeof(char));
while (1)
        {
            buf = fgetc(f);
            UKASATEL->next->price = realloc(UKASATEL->next->price, (sizest +
1) * sizeof(char));
if ((UKASATEL->next->price[sizest] = buf) == '\n')
break;
                sizest++;
            }
            UKASATEL->next->price[sizest] = '\0';
//////////
            sizest = 0;
            UKASATEL->next->kolvo = calloc(1, sizeof(char));

```

```

while (1)
{
    buf = fgetc(f);
    UKASATEL->next->kolvo = realloc(UKASATEL->next->kolvo, (sizest +
1) * sizeof(char));
    if ((UKASATEL->next->kolvo[sizest] = buf) == '\n')
break;
    sizest++;
}
UKASATEL->next->kolvo[sizest] = '\0';
////////////////////////////////////////
sizest = 0;
UKASATEL->next->sum = calloc(1, sizeof(char));
while (1)
{
    buf = fgetc(f);
    UKASATEL->next->sum = realloc(UKASATEL->next->sum, (sizest + 1) *
sizeof(char));
    if ((UKASATEL->next->sum[sizest] = buf) == '\n')
break;
    sizest++;
}
UKASATEL->next->sum[sizest] = '\0';

UKASATEL->next->next = NULL;

}

if (arr == NULL)
{
    free(arr);
arr = UKASATEL->next;
    free(UKASATEL);
    UKASATEL = arr;
}
else
    UKASATEL = UKASATEL->next;

if( j == 0)
    printf("Файл успешно прочтён\n\n");
    j++;

}

fclose(f);
return arr;
}

```

## Приложение А.7 Файл «search.c»

```

#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include"add.h"
#include"print.h"
#include"search.h"
#include<string.h>

void Search(struct naklad* arr) {
    char buf;
    char* str;
    struct naklad* UKASATEL = arr;
    int i = 1, sizest = 0;
    if (UKASATEL == NULL)
    {
        printf("\n%s\n", "СТРУКТУРА ПУСТА");
    }
}

```

```

return;
}

    str = malloc(sizeof(char));
printf("Введите наименование товара: "); getchar();
str = calloc(1, sizeof(char));
while (1)
{
    buf = 'm';
    str = realloc(str, (sizest + 1) * sizeof(char));
    if ((str[sizest] = buf) == 'm')
break;
    sizest++;
}

    scanf("%s", str);
printf("Найденные записи\n");
do
{
    if (strcmp(UKASATEL->name, str) == 0)
        printf("№%d\t%s\t%s\t%s\t%s\t%s\t\n", i, UKASATEL->name, UKASATEL->edizm, UKASATEL->price, UKASATEL->kolvo, UKASATEL->sum);
    i++;
    UKASATEL = UKASATEL->next;
} while (UKASATEL->next != NULL);
}

```

## Приложение А.8 Файл «add.c»

```

#define log_info(M, ...) fprintf(\ stderr, "[INFO] (%s:%d) " M "\n", \ __FILE__,
__LINE__, ##__VA_ARGS__)
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include"add.h"

struct naklad* Add(struct naklad* arr) {
    struct naklad* UKASATEL = arr;
    struct naklad* UKASATEL1 = arr;
    if (arr == NULL)
    {
        arr = malloc(sizeof(struct naklad));
        arr->next = NULL;
        UKASATEL = arr;
        arr = NULL;
    }
    while (1)
    {
        if (UKASATEL->next != NULL)
            UKASATEL = UKASATEL->next;
        else break;
    }
    int b = 0, i, sizest = 0;
    char buf;
    UKASATEL->next = malloc (sizeof(struct naklad));
    printf("Введите наименование товара: "); getchar();
    UKASATEL->next->name = calloc(1, sizeof(char));
    while (1)
    {
        buf = fgetc(stdin);
        UKASATEL->next->name = realloc(UKASATEL->next->name, (sizest + 1) *
sizeof(char));
        if ((UKASATEL->next->name[sizest] = buf) == '\n')
break;
        sizest++;
    }
}

```

```

        UKASATEL->next->name[sizest] = '\0';

//////////
        printf("Введите единицу измерения товара: ");
        UKASATEL->next->edizm = calloc(1, sizeof(char));
        sizest = 0;
        while (1)
        {
            buf = fgetc(stdin);
            UKASATEL->next->edizm = realloc(UKASATEL->next->edizm, (sizest + 1) *
sizeof(char));
            if ((UKASATEL->next->edizm[sizest] = buf) == '\n')
                break;
            sizest++;
        }
        UKASATEL->next->edizm[sizest] = '\0';
//////////
        do {
            b = 1;
            printf("Введите цену товара: ");
            sizest = 0;
            UKASATEL->next->price = calloc(1, sizeof(char));
            while (1)
            {
                buf = fgetc(stdin);
                UKASATEL->next->price = realloc(UKASATEL->next-> price, (sizest + 1) *
sizeof(char));
                if ((UKASATEL->next->price[sizest] = buf) == '\n')
                    break;
                sizest++;
            }
            UKASATEL->next->price[sizest] = '\0';
            for (int i = 0; i < sizest; i++)
                if (UKASATEL->next->price[i] == '+' || UKASATEL->next->price[i] == '(' ||
UKASATEL->next->price[i] == ')' || (UKASATEL->next->price[i] >= '0' && UKASATEL-
>next->price[i] <= '9'))
                    b = 0;
            if (b == 1)
                printf("Некорректный ввод, повторите ввод\n");
        } while (b == 1);
//////////
        do {
            b = 1;
            printf("Введите количество товара: ");
            sizest = 0;
            UKASATEL->next->kolvo = calloc(1, sizeof(char));
            while (1)
            {
                buf = fgetc(stdin);
                UKASATEL->next->kolvo = realloc(UKASATEL->next->kolvo, (sizest + 1) *
sizeof(char));
                if ((UKASATEL->next->kolvo[sizest] = buf) == '\n')
                    break;
                sizest++;
            }
            UKASATEL->next->kolvo[sizest] = '\0';
            for (int i = 0; i < sizest; i++)
                if (UKASATEL->next->kolvo[i] == '+' || UKASATEL->next->kolvo[i] == '(' ||
UKASATEL->next->kolvo[i] == ')' || (UKASATEL->next->kolvo[i] >= '0' && UKASATEL-
>next->kolvo[i] <= '9'))
                    b = 0;
            if (b == 1)
                printf("Некорректный ввод, повторите ввод\n");
        } while (b == 1);

        UKASATEL->next->sum = malloc(sizest);

```

```
    sprintf(UKASATEL->next->sum, "%d", atoi(UKASATEL->next->kolvo) *  
    atoi(UKASATEL->next->price));
```

```
    UKASATEL->next->next = NULL;  
    if (arr == NULL)  
    {  
        arr = UKASATEL->next;  
        free(UKASATEL);  
    }  
    printf("\n");  
    return arr;  
}
```

## Приложение В. Результат работы программы

### Приложение В.1



## Приложение В.2

```
1 - добавить товар
2 - удалить товар
3 - вывод товаров
4 - сохранить в файл
5 - загрузить из файла
6 - удалить файл
7 - поиск по имени
3

№1      хлеб   шт      30      40      1200
№2      яблоки кг      30      43      1290
№3      молоко литр  56      65      3640
Продолжить работу программы? (0 - Да, 1 - Нет)0

1 - добавить товар
2 - удалить товар
3 - вывод товаров
4 - сохранить в файл
5 - загрузить из файла
6 - удалить файл
7 - поиск по имени
7

Введите наименование товара: хлеб
Найденные записи
№1      хлеб   шт      30      40      1200
Продолжить работу программы? (0 - Да, 1 - Нет)0

1 - добавить товар
2 - удалить товар
3 - вывод товаров
4 - сохранить в файл
5 - загрузить из файла
6 - удалить файл
7 - поиск по имени
4

Успешно записано в файл
```

## Приложение С. Результат работы программы

Файл «tet.txt»

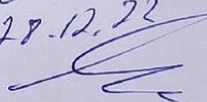
хлеб  
шт  
30  
40  
1200  
молоко  
литр  
55  
66  
3630  
груша  
кг  
67  
36  
2412  
ыкпкrrкrrфкфпф  
qqq www  
33  
33  
1089



Министерство образования Российской Федерации  
Пензенский государственный университет  
Кафедра «Вычислительная техника»

### ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе  
по курсу «Программирование»  
на тему «Разработка программы сложной структуры методом нисходящего  
программирования»

ОТЛ.  
28.12.22  


Выполнил: Митрошин М.А.  
Студент группы 21BO1  
Принял: Митрохин М.А.  
Зав. Кафедрой ВТ

Пенза, 2022

**ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет Вычислительной техники

Кафедра «Вычислительная техника»

«УТВЕРЖДАЮ»

Зав. кафедрой ВТ

Митрохин М. А.

«\_\_\_» \_\_\_\_\_ 2022 г

**ЗАДАНИЕ**

**на курсовое проектирование по курсу**

Программирование

Студенту Митрошину Максиму Андреевичу

Группа 21BO1

Тема проекта: Разработка программы сложной структуры

методом нисходящего программирования

**Исходные данные (технические требования) на проектирование**

Наименование программы: База данных «Накладная».

База данных включает следующие поля:

Имя товара, единица измерения, цена, количество, сумма.

**Задание.**

Программа должна выполнять следующие функции:

- Создание новой базы данных.
- Добавление/удаление записей, их редактирование.
- Поиск и сортировка записей.
- Сохранение базы данных в файл и удаление этого файла.
- Загрузка ранее сохраненной базы данных из файла.
- Суммирование количества товара при добавлении записи с идентичным названием.

**Обязательные требования к программе:**

1. Многомодульность.
2. Использование сложных типов данных — списков, структур, файлов.
3. Режим работы видеосистемы — текстовый/графический.
4. Устройство ввода информации — клавиатура, мышь.
5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
6. Наличие заставки.
7. Операционная система — Windows.
8. Система программирования — C/C++ на базе среды Microsoft Visual Studio 2019.
9. Язык программирования — Си.

Объем работы по курсу  
1. Расчетная часть  
Разработка программы.

2. Графическая часть  
Структура модулей программы, схема алгоритма программы.

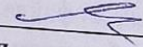
3. Экспериментальная часть  
Отладка программы.

**Срок выполнения проекта по разделам**

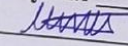
1. В соответствии с графиком выполнения курсового проекта.
- 2.
- 3.

Дата выдачи задания «1» октября

Дата защиты проекта «28» 12

Руководитель  Митрохин М.А.

Задание получил 1 «октября»

Студент  Митрошин Максим Андреевич

Введ

1. Г

2. Ве

3. От

4. От

5. От

5.1 Р

6. Ру

Закл

Спис

Прил

Прил

Прил