

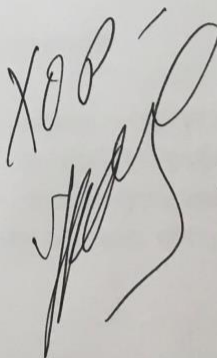
Министерство образования Российской Федерации
Пензенский государственный университет
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе
по курсу «Программирование»
на тему «Разработка программы сложной структуры методом многомодульного
программирования»
наименование программы «База данных. Адресная книга»

Выполнил студент группы 21ВВ4:

Куряев Ю.Р.



Принял:

Слепцов Н.В.

Пенза 2022

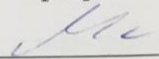
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет Вычислительной техники

Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ



М.А. Митрохин

« ____ » _____ 20__ г

ЗАДАНИЕ

на курсовое проектирование по курсу

Программирование

Студенту Куряеву Юнусу Рифатовичу Группа 21ВВ4

Тема проекта: *Разработка программы сложной структуры методом нисходящего программирования. База данных. Адресная книга*

Исходные данные: (технические требования) на проектирование
Добавление, удаление, редактирование, поиск, фильтрация, сортировка записей - Создание, удаление базы данных.

Обязательные требования к программе:

1. Многомодульности.
2. Использование сложных типов данных – структур, списков, 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 MS Windows.
8. Язык программирования - Си и Ассемблер.
9. Среда разработки ИО – Microsoft Visual Studio.

Объем работы по курсу

1. Расчетная часть

Разработка программы.

2. Графическая часть

Схема данных, схема ресурсов системы, схема работы системы, иерархическая структура программы, схема взаимодействия программы.

3. Экспериментальная часть

Отладка программы.

Срок выполнения проекта по разделам

В соответствии с графиком.

Дата выдачи задания " 25 " февраля 2022

Дата защиты проекта " "

Руководитель _____

Задание получил " 25 " февраля _____ 2022г.

Студент Куроев Ильяс Рифатоввич.

Содержание

Введение	4
1. Постановка задачи.....	5
2. Выбор решения.....	6
2.1. Определение необходимых модулей программы.....	6
2.2. Определение структуры базы данных	7
3. Описание разработки программы.....	7
4. Отладка и тестирование.....	11
5. Описание программы.....	11
6. Руководство пользователя	14
Заключение.....	17
Список используемых источников	18
Приложение	19
Листинг.....	19

Введение

База данных является совокупностью связанных данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования, независимая от других программ.

Задачей моей курсовой работы является разработка базы данных «Адресная книга», которая предполагает создание программы на языке Си с использованием класса статических массивов структур.

Структуры.

Структура представляет собой составной тип данных, который получается путем объединения компонентов, принадлежащих к другим типам данных. Структура предназначена для объединения нескольких переменных в один тип данных.

Существуют различные методы программирования для разработки программы. В данном случае, при создании курсовой работы использовался метод многомодульного программирования.

Использование нескольких модулей при написании программы увеличивает скорость трансляции и, в конечном итоге, скорость написания программы.

1. Постановка задачи.

Необходимо было разработать программу «База данных. Адресная книга». В данной программе информация структурирована по следующим полям: «ФИО», «Год рождения», «адрес», «индекс», «телефон». Для реализации данной программы необходимо определить структуру базы данных, размер, размер каждой записи базы данных.

Программа должна позволять пользователю осуществлять обработку данных, поэтому в программе должны присутствовать следующие функции:

- Создание и удаление базы данных
- Добавление записей
- Удаление записей
- Поиск записей
- Редактирование записей
- Фильтрация по определенному критерию и сортировка записей
- Сохранение текущей базы данных в файл
- Чтение базы данных из файла

Для того, чтобы реализовать вышеперечисленных требований необходимо разработать интуитивно понятный интерфейс пользователя, определить названия пунктов меню, определить тип интерфейса, продумать навигацию по базе данных, изучить функции для работы с файлами. Говоря об интерфейсе, нельзя не упомянуть о необходимости наличия заставки. Заставка-приветствие должна сообщать пользователю достаточную информацию о программе.

В программе должны использоваться сложные типы данных, такие как: структуры, массивы, списки, файлы. Это нужно для структурирования базы данных, для обработки данных в коде программы, для просмотра базы данных, с использованием файла, не используя запуск программы. Устройствами ввода являются клавиатура и компьютерная мышь. Навигация по пунктам меню и данные вводятся посредством клавиатуры. Программа должна быть разработана для работы в операционной системе MS Windows. Средой разработки является Microsoft Visual Studio 2010. В качестве языков программирования используются Си и Ассемблер.

2. Выбор решения.

2.1. Определение необходимых модулей программы.

Разработанная программа состоит из следующих модулей

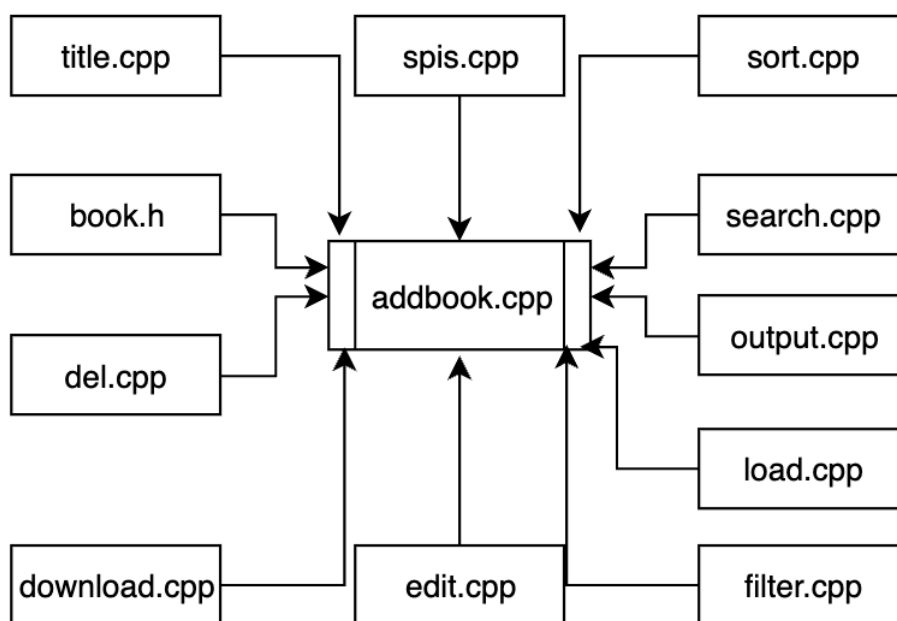


Рисунок 1 – Структура модулей программы

Файлы: «del.cpp», «edit.cpp», «download.cpp», «filter.cpp», «load.cpp», «addbook.cpp», «output.cpp», «search.cpp», «sort.cpp», «spis.cpp», «add.cpp» необходимы для включения функций обработки данных. Данные файлы содержат необходимые функции чтения, поиска, редактирование, добавления новых записей, создания, сортировки, фильтрации, вывода на экран базы данных и функцию сохранения результата с загрузкой из базы обработки базы данных. Файл addbook.cpp является связующим звеном, реализующим вышеописанные функции. В нем строится работа всех функции, навигационного меню и его подпунктов. Файле book.h является заголовком(библиотекой) храня в себе название процедур и функций, а также хранит в себе структуру базы данных.

Все вышеперечисленное удовлетворяет требованию модульности программы данной курсовой работы. Управление приложением осуществляется с помощью клавиатуры. При запуске программы выводится заставка-приветствие, после чего программа будет готова к дальнейшему исполнению. Программа предназначена для работы с базой данных, база данных находится в файле, файл так же можно создать самому. Открытие и сохранение выходного файла выполняется средствами данной

программы. Программа дает возможность просматривать содержимое базы данных, искать данные, сортировать их. Данные из базы отображаются в программе в виде таблицы из пяти столбцов «ФИО», «Год рождения», «Индекс», «Адрес», «Телефон».

2.2 Определение структуры базы данных

```
typedef struct addbook
{
    char fio[32],
        year[32],
        adres[32],
        index[32],
        phone[32];
    addbook* next;
};
```

Поля fio, year, adres, index, phone представляют собой символьную последовательность, имеющую ограничение в 32 символа.

Выбранная структура проста и соответствует всем требованиям программы.

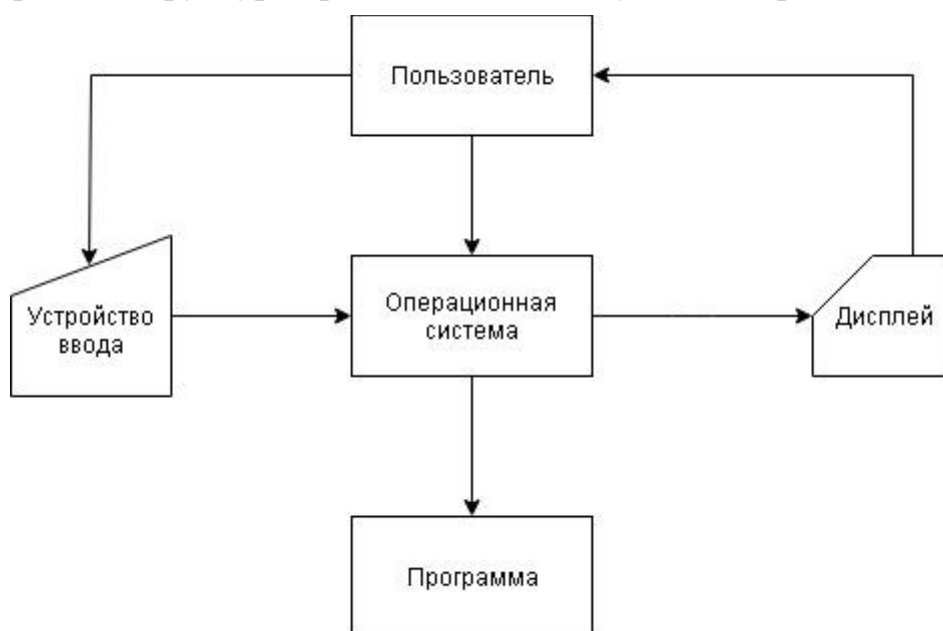


Рисунок 2. - Схема данных

3. Описание разработки программы.

В качестве среды программирования был выбран программный продукт Microsoft Visual studio 2010.

Интерфейс базы данных представляет собой меню. Перемещение по пунктам меню осуществляется при помощи цифр от 1 до 11, вводимых с клавиатуры.

При создании базы данных, после ввода каждого узла динамического списка будет выводиться сообщение о прекращении создания базы данных. Прекращение создания осуществляется вводом с клавиатуры '*'.

При выборе пункта меню «Создание базы данных», создается база данных, в навигационном меню предусмотрен пункт для обновления базы данных «Добавить». Данный пункт позволяет добавить новый узел в конец существующей базы данных.

Для корректировки предусмотрена функция удаления последнего узла, находящаяся в пункте «Удаление» навигационного меню.

Также в программе описаны такие пункты меню, как «Фильтрация», «Поиск». Первый пункт осуществляет возможность отбора данных по определенному критерию. Второй пункт дает пользователю возможность найти в базе данных необходимую информацию по определенному слову.

Пункт меню «Загрузка» предоставляет возможность представить базу данных в текстовом формате, создавая для этого файл. Также пользователь вводит нужное ему имя файла и разрешение, после чего произойдет его создание и запись базы данных.

Пункт меню «Вывод базы данных» позволяет в любой момент использования программы выбрать его и тем самым просмотреть содержимое существующей базы данных.

Пункт меню «Получить из файла» позволяет загрузить базу данных из файла в память программы.

Пункт меню «Редактирование» позволяет вносить изменения в уже существующую базу данных.

Пункт меню «Сортировка» позволяет сортировать базу данных по Названию или году выпуска.

После завершения работы с базой данных для выхода из программы существует пункт меню «Выход». После выполнения данного пункта существующая база данных будет удалена из памяти программы.

Разработка программы началась с файла **spis.cpp**, в котором первой была написана основная функция для базы данных – это функция создания базы данных **spisok**. Для быстрого поиска данных, находящихся в базе данных, была создана функция **search()**, которая осуществляет поиск по всем полям. Затем были описаны функции сохранения базы данных в файл – **load()**. Также для функции **del()** позволяет удалить последний узел динамического списка. Функция **output()** отвечает за корректный вывод базы данных на экран. Функция **filter()** позволяет отфильтровать базу данных по введенному фильтру по всем возможным полям. Функция **edit()** позволяет редактировать базу данных. Функция **sort()** позволяет сортировать базу данных по двум полям Название и год выпуска. Функциями информации и навигационного меню является функция **main()**. Функция **main()** является главной функцией проекта и находится в файле **addbookccp**. Главной ее задачей является осуществление связи между информационными функциями и функциями, выполняющие задачи, указанные в навигационном меню и вывод навигационного меню. На экран печатается навигационное меню, после чего запрашивается ввод цифры от 1 до 11. Каждой цифре соответствует определенный пункт меню. После выбора пункта меню выбирается подпункт меню, затем вызывается нужная функция. Схема программы функции **main()** представлена на рисунке ниже.

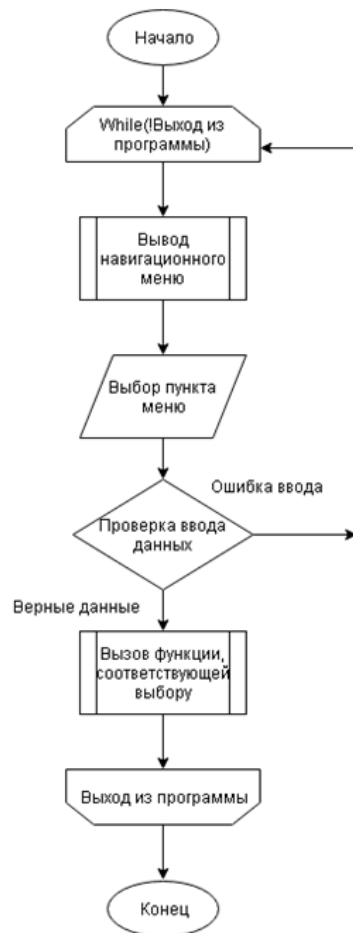


Рисунок 3. - Схема программы функции main()

Как только главная функция обнаружила событие от клавиатуры или мыши, вызывается функция обработки события. В зависимости от активного элемента выполняются те или иные действия. Если события принадлежат меню, то выполняется перемещение выделенного пункта меню. Если выбрано меню, то вызываются функции реализации меню. В функциях реализации меню находится свой цикл проверки событий клавиатуры или мыши. На выходе из функций реализации меню определяется, как ой из пунктов меню был выбран. Алгоритм представлен на рисунке 4.

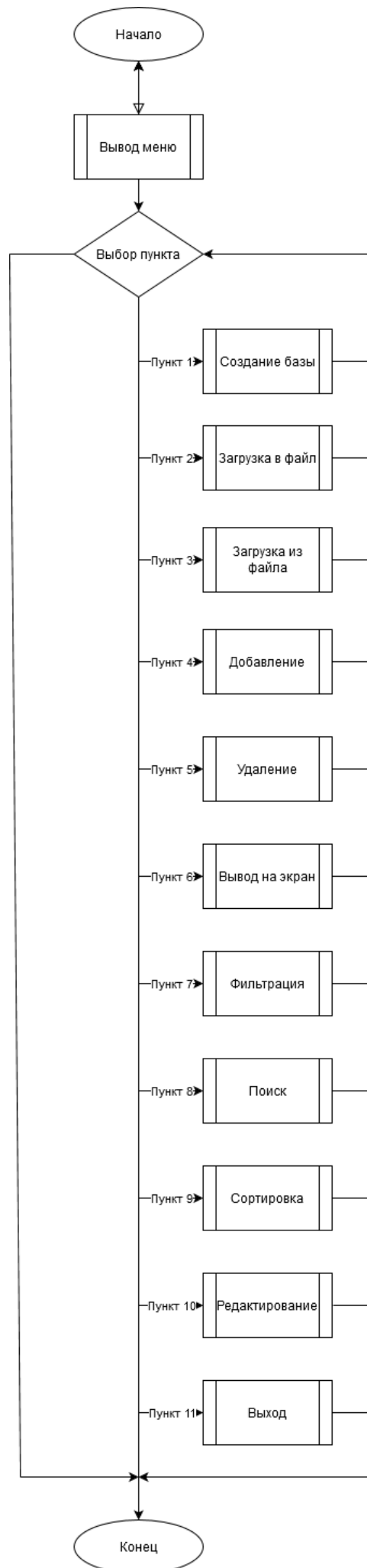


Рисунок 4. - Схема навигационного меню addbook.cpp

4. Отладка и тестирование.

При создании проекта использовалась программа Microsoft Visual Studio 2010. Она имеет все необходимые для разработки и отладки программы. Для отладки программы использовались: точки остановок, анализ содержимого локальных и глобальных переменных, пошаговое выполнение кода.

5. Описание программы.

После запуска программы появляется навигационное меню, содержащее пункты, которые имеют определенную нумерацию. Для выбора необходимого пункта меню доступен ввод цифры, которой соответствует определенный пункт меню. После ввода цифры осуществляется выполнение функции, то есть ожидается выполнение определенного действия, либо выводится подменю, в котором нужно выбрать необходимое действие.

Структура главного меню:

№	Опция	Действия программы
1	Создание базы данных	Производится создание новой базы данных
2	Загрузка	База загружается в файл
3	Получить из файла	База загружается из файла
4	Добавить	Добавление новых строк
5	Удаление	Удаление последней строки
6	Вывод базы данных	Вывод базы данных
7	Поиск	Находит в базе данных необходимую информацию по определенному слову.

8	Фильтрация	Отбор записей по определенному критерию
9	Сортировка	Данный пункт осуществляет возможность отбора данных по определенному критерию.
10	Редактирование	Ввод данных для редактирования записи
11	Выход	Запрос на подтверждение выхода

Иерархическая структура программы представлена на рисунке 5.

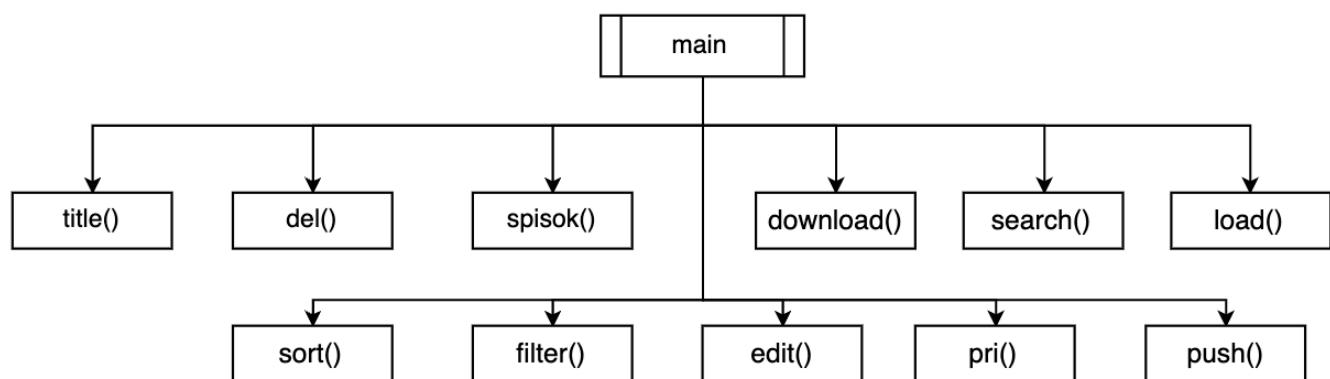


Рисунок 5 – Иерархическая структура программы

Блок схема для функции output (вывод базы данных на экран).

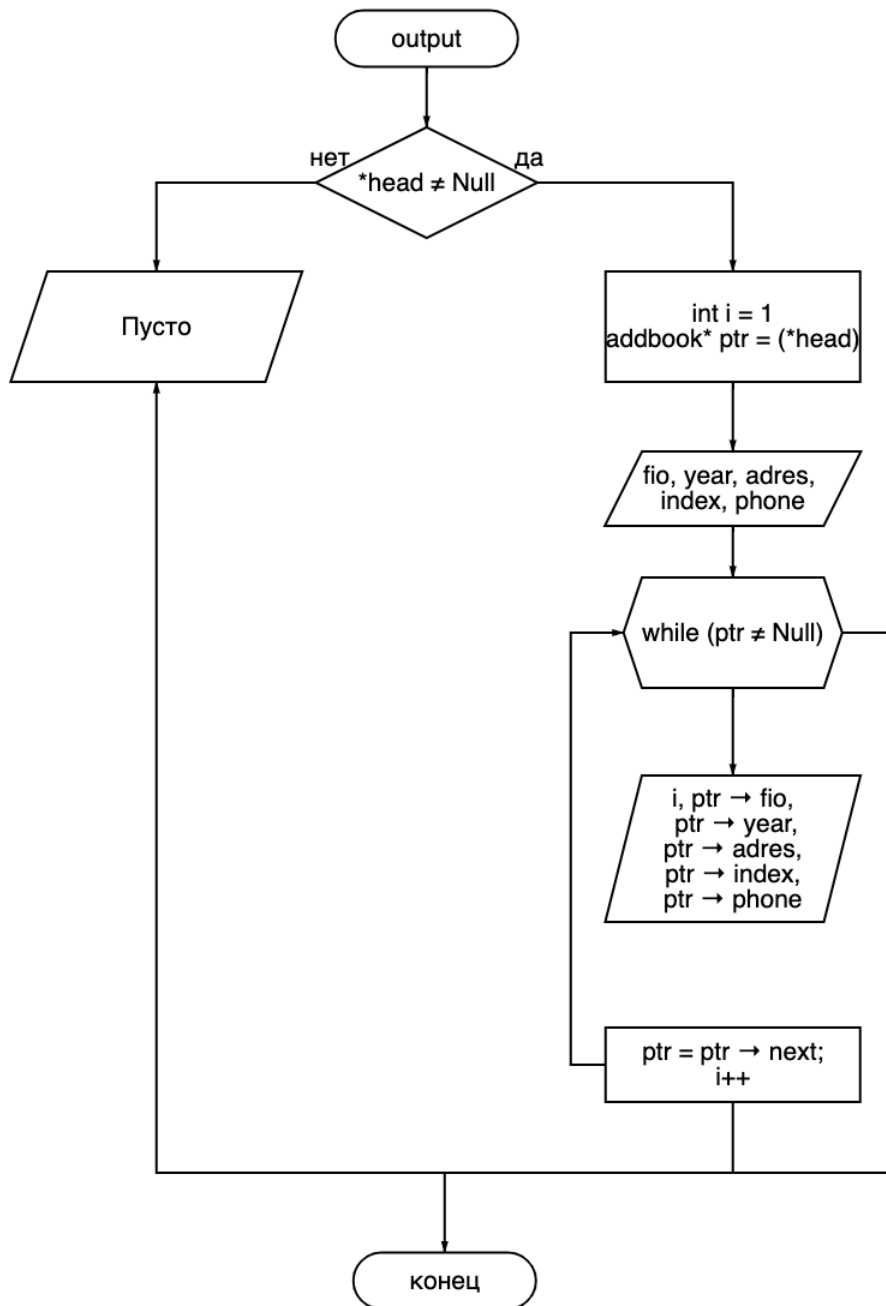


Рисунок 6 – Схема программирования для функции output

6. Руководство пользователя

При запуске программы у нас появляется заставка-приветствие.

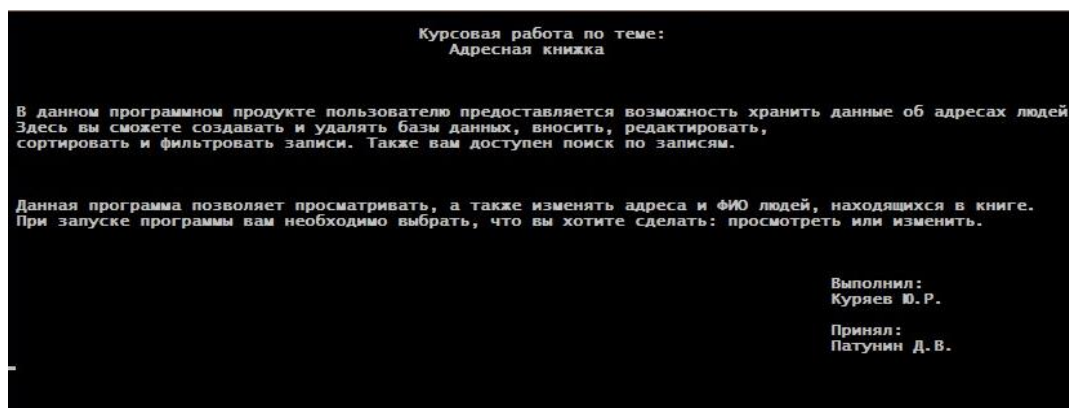


Рисунок 6. – заставка-приветствие

Для чтения и вывода на экран существующего списка необходимо выбрать пункт меню «Получить из файла », либо можно создать новую базу данных, выбрав пункт меню «Создание базы данных».

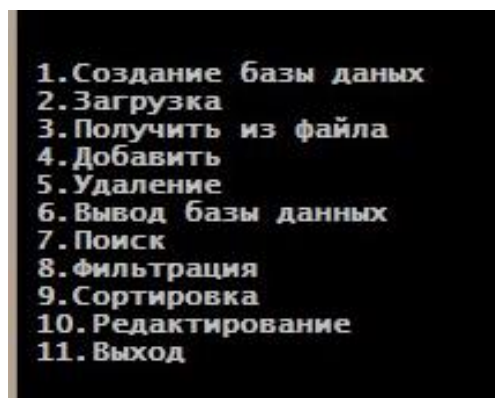


Рисунок 7. - Меню программы

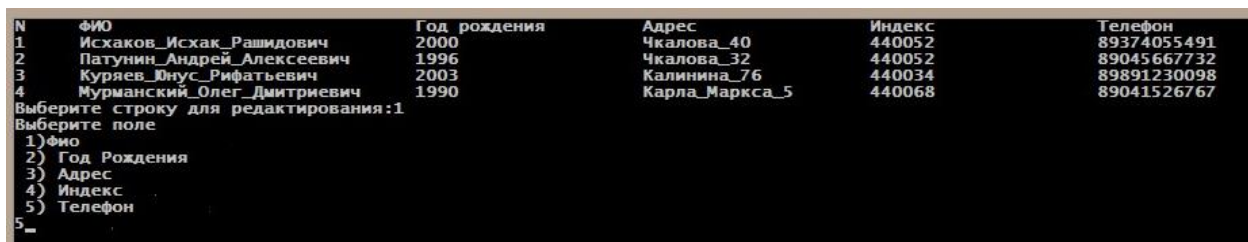
После этого произойдет чтение и вывод базы данных.

N	ФИО	Год рождения	Адрес	Индекс	Телефон
1	Мурманский Олег Дмитриевич	1990	Карла Маркса_5	440068	89041526767
2	Куряев Юнус Рифатьевич	2003	Калинина_76	440034	89891230098
3	Патунин Андрей Алексеевич	1996	Чкалова_32	440052	89045667732
4	Исхаков Исхак Рашидович	2000	Чкалова_40	440052	89374055491

Для продолжения нажмите любую клавишу . . . _

Рисунок 8. – Вывод базы данных

Для редактирования базы данных следует выбрать пункт 10.



N	ФИО	Год рождения	Адрес	Индекс	Телефон
1	Исхаков_Исхак_Рашидович	2000	Чкалова_40	440052	89374055491
2	Патунин_Андрей_Алексеевич	1996	Чкалова_32	440052	89045667732
3	Куряев_Юнус_Рифатьевич	2003	Калинина_76	440034	89891230098
4	Мурманский_Олег_Дмитриевич	1990	Карла_Маркса_5	440068	89041526767

Выберите строку для редактирования:1
Выберите поле
1)ФИО
2) Год Рождения
3) Адрес
4) Индекс
5) Телефон
5_

Рисунок 9. – Редактирование

После окончания ввода следует нажать клавишу Enter. В случае успешного выполнения операции программа выйдет в главное меню.

Для создания файла и записи в него созданной базы данных предназначен пункт меню 2

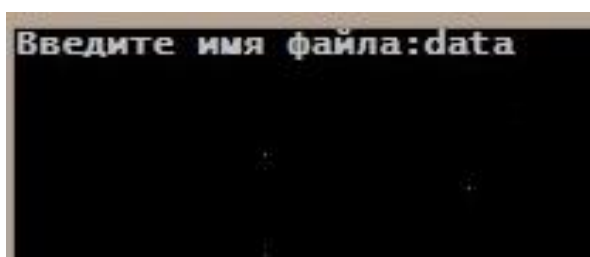
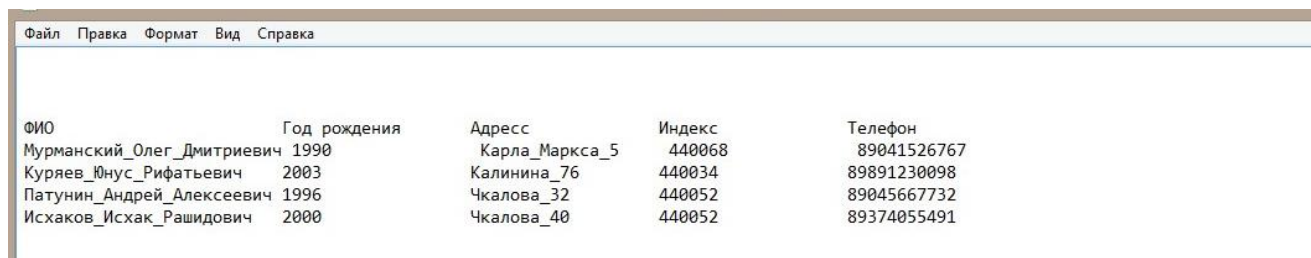


Рисунок 10. - Запись базы данных в файл

Результат будет записан в файле data



ФИО	Год рождения	Адрес	Индекс	Телефон
Мурманский_Олег_Дмитриевич	1990	Карла_Маркса_5	440068	89041526767
Куряев_Юнус_Рифатьевич	2003	Калинина_76	440034	89891230098
Патунин_Андрей_Алексеевич	1996	Чкалова_32	440052	89045667732
Исхаков_Исхак_Рашидович	2000	Чкалова_40	440052	89374055491

Рисунок 11. - Запись базы данных в файл data

Для получения базы данных из файла нужно воспользоваться пунктом 3 и ввести название файла, из которого будет получена база данных.

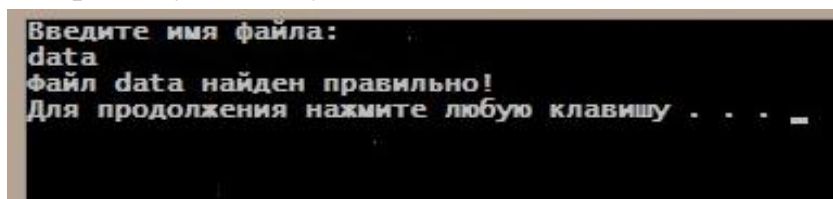


Рисунок 12. - Сообщение об успешном создании файла

Чтобы отфильтровать записи по значениям одного из полей в меню создан пункт 7. После выбора данного пункта появляется подменю, которое предлагает фильтрацию по доступным полям. Далее запрашивается выбор пункта в подменю и после этого выводится результат. Выбираем фильтрацию по жанру



Рисунок 13. – Фильтрация

Для удаления последнего узла базы данных предназначен пункт меню под номером 5.

Если узел необходимо добавить, то следует воспользоваться 4-ым пунктом меню. Этот пункт позволяет добавлять узел в конец базы данных.

Пункт 8 в главное меню осуществляет поиск по слову. Пользователь должен ввести слово, после чего выведется результат.

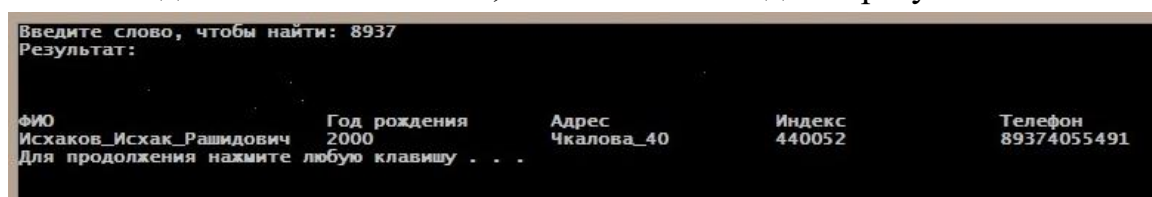
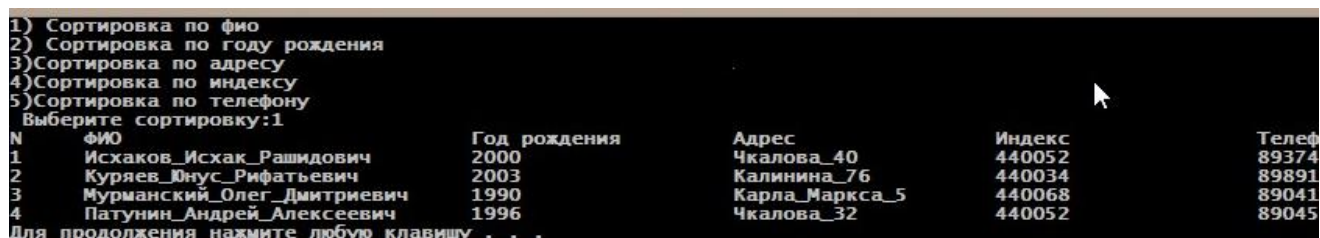


Рисунок 14. - Поиск по слову

Пункт 10 в главном меню позволяет редактировать записи в базе данных.

Пункт меню под номером 9 выполняет сортировку некоторым полям.



1) Сортировка по фео
2) Сортировка по году рождения
3) Сортировка по адресу
4) Сортировка по индексу
5) Сортировка по телефону
Выберите сортировку:1

N	ФИО	Год рождения	Адрес	Индекс	Телеф
1	Исхаков Исхак Рашидович	2000	Чкалова_40	440052	89374
2	Куряев Юнус Рифатьевич	2003	Калинина_76	440034	89891
3	Мурманский Олег Дмитриевич	1990	Карла_Маркса_5	440068	89041
4	Патунин Андрей Алексеевич	1996	Чкалова_32	440052	89045

Для продолжения нажмите любую клавишу . . .

Рисунок 16. – Список после сортировки по ФИО

Последний пункт 11 осуществляет выход из программы.

Заключение

При выполнении данной курсовой работы были получены навыки разработки многомодульных программ. Были освоены приемы создания псевдографического интерфейса, изучены функции работы с консолью, способы обработки событий от клавиатуры. Усвоены механизмы реализации меню. Изучены функции работы с файлами. Получены базовые навыки программирования на языке Си. Изучены основные возможности среды программирования Visual Studio 2017. Получены навыки отладки и тестирования программ.

В процессе выполнения этой лабораторной работы была создана программа «База данных. Адресная книга.», которая предназначена для хранения, обработки, редактирования, фильтрации и сортировки данных о ФИО, годе рождения, индексу, номеру телефона и адресу.

Список используемых источников

1. <http://cppstudio.com>
2. <https://cyberforum.ru>
3. Герберт Шилдт. Полный справочник по С.
4. <http://programmersforum.ruc>
5. <https://prog-cpp.ru/cpp/>
6. https://cpp.com.ru/shildt_spr_po_c/

Приложение

Листинг

Файл book.h

```
#pragma once

typedef struct addbook
{
    char fio[32],
        year[32],
        adres[32],
        index[32],
        phone[32];

    addbook* next;
};

void sort(addbook** head);
void sort_fio(addbook** head);
void sort_year(addbook** head);
void sort_adres(addbook** head);
void sort_index(addbook** head);
void sort_phone(addbook** head);
void spisok(addbook** head);
void download(addbook** head);
void title();
void del(addbook** head);
void search(addbook** head);
void filter(addbook** head);
void edit(addbook** head);
void load(addbook** head);
void download(addbook** head);
void pri(addbook** head);
void push(addbook** head, char* fio, char* year, char* adres, char* index, char* phone);
```

Файл addbook.cpp (с ассемблерной вставкой)

```
#define _CRT_SECURE_NO_WARNINGS
#include "stdafx.h"
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <locale.h>
#include <io.h>
#include <Windows.h>
#include "book.h"

void main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    title();
    system("pause");
    int count;
    addbook* head = NULL;
    for (;;)
    {
        system("cls");
        char menu[] = "\n\n 1.Создание базы данных\n 2.Загрузка\n 3.Получить из файла\n";
```

8.Фильтрация\n" " 4.Добавить\n 5.Удаление\n 6.Вывод базы данных\n 7.Поиск\n " 9.Сортировка\n 10.Редактирование\n 11.Выход\n\n";

```
__asm
{
    xor eax, eax
    lea eax, menu
    push eax
    call dword ptr[printf]
    add esp, 4
}
printf("\n Введите номер операции:");
scanf("%d", &count);
```

```
switch (count)
{
case 1:
    system("cls");
    spisok(&head);
    break;
case 2:
    system("cls");
    load(&head);
    break;
case 3:
    system("cls");
    download(&head);
    system("pause");
    break;
case 4:
    system("cls");
    spisok(&head);
    system("pause");
    break;
case 5:
    system("cls");
    del(&head);
    system("pause");
    break;
case 6:
    system("cls");
    fflush(stdin);
    pri(&head);
    system("pause");
    break;
case 7:
    system("cls");
    search(&head);
    system("pause");
    break;
case 8:
    system("cls");
    filter(&head);
    system("pause");
    break;
case 9:
    system("cls");
    sort(&head);
    system("pause");
    break;
case 10:
    system("cls");
    edit(&head);
    pri(&head);
    break;
case 11:
    exit(0);
    break;
```

```

    }
}

struct addbook* node; //удаление списка{
while (head != NULL)
{
    node = head;
    head = head->next;
    free(node); // удаление списка}
}
}

```

Файл download.cpp

```

#include "stdafx.h"
#include "book.h"

void download(addbook** head)
{
    int i;
    FILE* S;
    char name[32], fio[32], year[32], adres[32], index[32], phone[32];
    int fd;
    *head = NULL;
    fprintf(stdout, "%s", "Введите имя файла:\n");
    scanf("%s", &name);
    while ((S = fopen(name, "r+")) == (NULL))
    {
        fprintf(stdout, "%s", "Файл не найден!\nПопробуйте еще раз:");
        scanf("%s", &name);
    }
    fprintf(stdout, "Файл %s найден правильно!\n", name);
    if (!feof(S))
    {
        fseek(S, 100, SEEK_SET);
        while (!feof(S))
        {
            fscanf(S, "%s", &fio);
            fscanf(S, "%s", &year);
            fscanf(S, "%s", &adres);
            fscanf(S, "%s", &index);
            fscanf(S, "%s", &phone);
            if (!feof(S))
                push(head, fio, year, adres, index, phone);
        }
    }
    else
    {
        printf("\nПУСТО!\n\n");
    }
    fclose(S);
}

```

Файл load.cpp

```

#include "stdafx.h"
#include "book.h"

void load(addbook** head)
{
    FILE* S;
    addbook* ptr = (*head);
    char name[32];
    fflush(stdin);
    fprintf(stdout, "%s", "Введите имя файла:");
    scanf("%s", &name);
    while ((S = fopen(name, "w")) == (NULL))
    {

```

```

        fprintf(stdout, "%s", "Файл не найден!\n");
        scanf("%s", &name);
    }
    fprintf(stdout, "Файл %s был найден успешно!\n", name);
    if ((*head) != NULL)
    {
        addbook* ptr = *head;
        fprintf(S, "\n\n%-25s %-18s %-18s %-18s %-20s\n", "ФИО", "Год рождения", "Адресс",
"Индекс", "Телефон");
        while (ptr != NULL)
        {
            fprintf(S, "%-25s %-18s %-18s %-18s %-20s\n", ptr->fio, ptr->year, ptr->adres,
ptr->index, ptr->phone);
            ptr = ptr->next;
        }
    }
    else
    {
        printf("ПУСТО\n\n");
    }
    fclose(S);
}

```

Файл search.cpp

```

#include "stdafx.h"
#include "book.h"
#include <string.h>

void search(addbook** head)
{
    char name[16];
    addbook* ptr = (*head);
    fflush(stdin);
    printf("Введите слово, чтобы найти: ");
    gets(name);
    if (ptr == NULL)
        printf("ПУСТО");
    else {
        printf("Результат:\n");
        printf("\n\n%-25s %-18s %-18s %-18s %-20s\n", "ФИО", "Год рождения", "Адрес",
"Индекс", "Телефон");
        while (ptr != NULL)
        {
            if (strstr(ptr->fio, name) != NULL)
                printf("%-25s %-18s %-18s %-18s %-20s\n", ptr->fio, ptr->year, ptr->adres, ptr-
>index, ptr->phone);
            if (strstr(ptr->year, name) != NULL)
                printf("%-25s %-18s %-18s %-18s %-20s\n", ptr->fio, ptr->year, ptr->adres, ptr-
>index, ptr->phone);
            if (strstr(ptr->adres, name) != NULL)
                printf("%-25s %-18s %-18s %-18s %-20s\n", ptr->fio, ptr->year, ptr->adres, ptr-
>index, ptr->phone);
            if (strstr(ptr->index, name) != NULL)
                printf("%-25s %-18s %-18s %-18s %-20s\n", ptr->fio, ptr->year, ptr->adres, ptr-
>index, ptr->phone);
            if (strstr(ptr->phone, name) != NULL)
                printf("%-25s %-18s %-18s %-18s %-20s\n", ptr->fio, ptr->year, ptr->adres, ptr-
>index, ptr->phone);
            ptr = ptr->next;
        }
    }
}

```

Файл title.cpp

```

#include "stdafx.h"
#include "book.h"
#include <iostream>
#include <conio.h>

void title()
{
    const char name[] = "\n
                                Курсовая работа по теме:\n"
                                Адресная книжка\n\n\n\n";
    const char opis[] =
        " В данном программном продукте пользователю предоставляется возможность хранить данные об
адресах людей.\n"
        " Здесь вы сможете создавать и удалять базы данных, вносить, редактировать,\n"
        " сортировать и фильтровать записи. Также вам доступен поиск по записям.\n\n\n\n"
        " Данная программа позволяет просматривать, а также изменять адреса и ФИО людей, находящихся
в книге.\n"
        " При запуске программы вам необходимо выбрать, что вы хотите сделать: просмотреть или
изменить.\n\n\n\n";
    const char rez[] = "
                                Выполнил:\n"
                                Куряев Ю.Р.\n\n"
                                Принял:\n"
                                Патунин Д.В.\n";

    std::cout << name << opis << rez;
    _getch();
}

```

Файл output.cpp

```

void pri(game_list** head)
{
    if (*head != NULL)
    {
        int i = 1;
        game_list* ptr = (*head);
        ptr = (*head);
        printf("N      %-30s %-20s %-20s %-20s %-20s\n", "Name", "Developer", "Genre",
"Salles", "Year");
        while (ptr != NULL)
        {
            printf("%-5d %-30s %-20s %-20s %-20s %-20s\n", i, ptr->nazvanie, ptr-
>developer, ptr->genre, ptr->sells, ptr->year);
            ptr = ptr->next;
            i++;
        }
    }
    else
    {
        printf("EMPTY");
    }
}

```

Файл sort.cpp

```

#define _CRT_SECURE_NO_WARNINGS
#include "stdafx.h"
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <locale.h>
#include <io.h>
#include <Windows.h>
#include "book.h"

void sort_phone(addbook** head)
{
    char fio[32], year[32], adres[32], index[32], phone[32];
    int size = 0;
    addbook* ptr = (*head);
    while (ptr != NULL)
    {

```

```

        ptr = ptr->next;
        size++;
    }
    ptr = (*head);
    for (int j = 1; j < size; j++)
    {
        ptr = (*head);
        for (int i = 0; i < size - j; i++)
        {
            if (atoi(ptr->phone) > atoi(ptr->next->phone))
            {
                strcpy(fio, ptr->fio);
                strcpy(year, ptr->year);
                strcpy(adres, ptr->adres);
                strcpy(index, ptr->index);
                strcpy(phone, ptr->phone);

                strcpy(ptr->fio, ptr->next->fio);
                strcpy(ptr->year, ptr->next->year);
                strcpy(ptr->adres, ptr->next->adres);
                strcpy(ptr->index, ptr->next->index);
                strcpy(ptr->phone, ptr->next->phone);

                strcpy(ptr->next->fio, fio);
                strcpy(ptr->next->year, year);
                strcpy(ptr->next->adres, adres);
                strcpy(ptr->next->index, index);
                strcpy(ptr->next->phone, phone);
            }
            ptr = ptr->next;
        }
    }
}

void sort_fio(addbook** head)
{
    char fio[32], year[32], adres[32], index[32], phone[32];
    int size = 0;
    addbook* ptr = (*head);
    while (ptr != NULL)
    {
        ptr = ptr->next;
        size++;
    }
    ptr = (*head);
    for (int j = 1; j < size; j++)
    {
        ptr = (*head);
        for (int i = 0; i < size - j; i++)
        {
            if (ptr->fio[0] > ptr->next->fio[0])
            {
                strcpy(fio, ptr->fio);
                strcpy(year, ptr->year);
                strcpy(adres, ptr->adres);
                strcpy(index, ptr->index);
                strcpy(phone, ptr->phone);

                strcpy(ptr->fio, ptr->next->fio);
                strcpy(ptr->year, ptr->next->year);
                strcpy(ptr->adres, ptr->next->adres);
                strcpy(ptr->index, ptr->next->index);
                strcpy(ptr->phone, ptr->next->phone);

                strcpy(ptr->next->fio, fio);
                strcpy(ptr->next->year, year);
                strcpy(ptr->next->adres, adres);
                strcpy(ptr->next->index, index);
                strcpy(ptr->next->phone, phone);
            }
        }
    }
}

```

```

        }
        ptr = ptr->next;
    }
}

void sort_index(addbook** head)
{
    char fio[32], year[32], adres[32], index[32], phone[32];
    int size = 0;
    addbook* ptr = (*head);
    while (ptr != NULL)
    {
        ptr = ptr->next;
        size++;
    }
    ptr = (*head);
    for (int j = 1; j < size; j++)
    {
        ptr = (*head);
        for (int i = 0; i < size - j; i++)
        {
            if (atoi(ptr->index) > atoi(ptr->next->index))
            {
                strcpy(fio, ptr->fio);
                strcpy(year, ptr->year);
                strcpy(adres, ptr->adres);
                strcpy(index, ptr->index);
                strcpy(phone, ptr->phone);

                strcpy(ptr->fio, ptr->next->fio);
                strcpy(ptr->year, ptr->next->year);
                strcpy(ptr->adres, ptr->next->adres);
                strcpy(ptr->index, ptr->next->index);
                strcpy(ptr->phone, ptr->next->phone);

                strcpy(ptr->next->fio, fio);
                strcpy(ptr->next->year, year);
                strcpy(ptr->next->adres, adres);
                strcpy(ptr->next->index, index);
                strcpy(ptr->next->phone, phone);
            }
            ptr = ptr->next;
        }
    }
}

```

```

void sort_adres(addbook** head)
{
    char fio[32], year[32], adres[32], index[32], phone[32];
    int size = 0;
    addbook* ptr = (*head);
    while (ptr != NULL)
    {
        ptr = ptr->next;
        size++;
    }
    ptr = (*head);
    for (int j = 1; j < size; j++)
    {
        ptr = (*head);
        for (int i = 0; i < size - j; i++)
        {
            if (ptr->adres[0] > ptr->next->adres[0])
            {
                strcpy(fio, ptr->fio);
                strcpy(year, ptr->year);
                strcpy(adres, ptr->adres);
                strcpy(index, ptr->index);
            }
        }
    }
}

```

```

        strcpy(phone, ptr->phone);

        strcpy(ptr->fio, ptr->next->fio);
        strcpy(ptr->year, ptr->next->year);
        strcpy(ptr->adres, ptr->next->adres);
        strcpy(ptr->index, ptr->next->index);
        strcpy(ptr->phone, ptr->next->phone);

        strcpy(ptr->next->fio, fio);
        strcpy(ptr->next->year, year);
        strcpy(ptr->next->adres, adres);
        strcpy(ptr->next->index, index);
        strcpy(ptr->next->phone, phone);
    }
    ptr = ptr->next;
}
}
}

```

```

void sort_year(addbook** head)
{
    char fio[32], year[32], adres[32], index[32], phone[32];
    int size = 0;
    addbook* ptr = (*head);
    while (ptr != NULL)
    {
        ptr = ptr->next;
        size++;
    }
    ptr = (*head);
    for (int j = 1; j < size; j++)
    {
        ptr = (*head);
        for (int i = 0; i < size - j; i++)
        {
            if (ptr->year[0] > ptr->next->year[0])
            {
                strcpy(fio, ptr->fio);
                strcpy(year, ptr->year);
                strcpy(adres, ptr->adres);
                strcpy(index, ptr->index);
                strcpy(phone, ptr->phone);

                strcpy(ptr->fio, ptr->next->fio);
                strcpy(ptr->year, ptr->next->year);
                strcpy(ptr->adres, ptr->next->adres);
                strcpy(ptr->index, ptr->next->index);
                strcpy(ptr->phone, ptr->next->phone);

                strcpy(ptr->next->fio, fio);
                strcpy(ptr->next->year, year);
                strcpy(ptr->next->adres, adres);
                strcpy(ptr->next->index, index);
                strcpy(ptr->next->phone, phone);
            }
            ptr = ptr->next;
        }
    }
}

```

```

void sort(addbook** head)
{
    int a;
    printf("1) Сортировка по фео\n2) Сортировка по году рождения\n3)Сортировка по
адресу\n4)Сортировка по индексу\n5)Сортировка по телефону\n Выберите сортировку:");
    scanf("%d", &a);
    switch (a)
    {
        case 1:

```

```

        sort_fio(head);
        pri(head);
        system("pause");
        break;
    case 2:
        sort_phone(head);
        pri(head);
        system("pause");
        break;
    case 3:
        sort_index(head);
        pri(head);
        system("pause");
        break;
    case 4:
        sort_year(head);
        pri(head);
        system("pause");
        break;
    case 5:
        sort_adres(head);
        pri(head);
        system("pause");
        break;
    default:
        break;
}
}

```

Файл edit.cpp

```

#include "stdafx.h"
#include "book.h"
#include <Windows.h>

void edit(addbook** head)
{
    addbook* ptr = (*head);
    int i = 0;
    char name[16];
    pri(head);
    if (ptr != NULL)
    {
        ptr = (*head);
        printf("Выберите строку для редактирования:");
        scanf("%d", &i);
        i--;
        for (; i > 0; i--)
        {
            ptr = ptr->next;
            if (ptr == NULL)
            {
                exit(-1);
                printf("Ошибка при вводе!");
            }
        }
        printf("Выберите поле \n 1)Фιο \n 2) Год Рождения \n 3) Адрес \n 4) Индекс \n 5)
Телефон \n");
        scanf("%d", &i);
        fflush(stdin);
        printf("Введите новое значение: ");
        gets(name);
        switch (i)
        {
            case 1:
                strcpy(ptr->fio, name);
                break;
            case 2:
                strcpy(ptr->year, name);

```

```

        break;
    case 3:
        strcpy(ptr->adres, name);
        break;
    case 4:
        strcpy(ptr->index, name);
        break;
    case 5:
        strcpy(ptr->phone, name);
        break;
    default: break;
}

}
else {
    printf("ПУСТО!");
}
}

```

Файл filter.cpp

```

#include "stdafx.h"
#include "book.h"
#include <string.h>
void filter(addbook** head)
{
    addbook* ptr = (*head);
    fflush(stdin);
    fflush(stdout);
    char number[16];
    int i;
    printf(" 1)Год рождения\n 2)Индекс\n Выберите критерий фильтрации:");
    scanf("%d", &i);
    fflush(stdin);
    switch (i)
    {
    case 1:
        fflush(stdin);
        printf("Введите год рождения:");
        gets(number);
        while (ptr != NULL)
        {
            if (strstr(ptr->year, number) != NULL)
                printf("%-32s %-32s %-32s %-32s %-32s\n", ptr->fio, ptr->year, ptr->adres, ptr->index, ptr->phone);
            ptr = ptr->next;
        }
        break;

    case 2:
        fflush(stdin);
        printf("Введите индекс:");
        gets(number);
        while (ptr != NULL)
        {
            if (strstr(ptr->index, number) != NULL)
                printf("%-32s %-32s %-32s %-32s %-32s\n", ptr->fio, ptr->year, ptr->adres, ptr->index, ptr->phone);
            ptr = ptr->next;
        }
        break;
    default: printf("Ошибка!");
    }
}

```

Файл del.cpp

```

#include "stdafx.h"
#include "book.h"
#include <Windows.h>

void del(addbook** head)

```

```

{
    system("cls");
    printf("\nВведите номер строки которую хотите удалить\n");
    int n;
    scanf("%d", &n);
    n--;

    if ((*head) != NULL)
    {
        if (n == 0) {
            addbook* prev = (*head);
            (*head) = (*head)->next;

            delete prev;
        }
        else {
            addbook* previous = (*head);
            for (int i = n-1; i > 0; i--) {
                previous = previous->next;
            }
            addbook* todelete = previous->next;
            previous->next = todelete->next;
            delete todelete;
        }
    }
    else
    {
        printf("\nСписок пуст!\n");
    }
}

```

Файл spis.cpp

```

#include "stdafx.h"
#include "book.h"
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <locale.h>
#include <io.h>
#include <Windows.h>

void spisok(addbook** head)
{
    fflush(stdin);
    char fio[32], year[32], adres[32], index[32], phone[32];
    printf("Введите Фιο:");
    gets(fio);
    for (;;)
    {
        fflush(stdin);
        printf("Введите Год рождения:");
        gets(year);
        printf("Введите Адрес:");
        gets(adres);
        printf("Введите Индекс:");
        gets(index);
        printf("Введите Телефон:");
        gets(phone);
        push(head, fio, year, adres, index, phone);
        printf("Для завершения введите: *\nВведите ФИО:");
        gets(fio);
        if (fio[0] == '*')
            break;
    }
}

void push(addbook** head, char* fio, char* year, char* adres, char* index, char* phone)
{

```

```
addbook* tmp;  
tmp = new addbook;  
strcpy_s(tmp->fio, fio);  
strcpy_s(tmp->year, year);  
strcpy_s(tmp->adres, adres);  
strcpy_s(tmp->index, index);  
strcpy_s(tmp->phone, phone);  
tmp->next = (*head);  
(*head) = tmp;
```

```
}
```