

Министерство науки и высшего образования РФ
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»

Выполнила студентка группы 21ВО2:

Земскова В.М.

Принял:

Митрохин М.А.

Пенза 2022

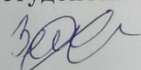
Министерство науки и высшего образования РФ
Пензенский государственный университет
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

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

кор.
28.12.22
М

Выполнила студентка группы 21ВО2:

 Земскова В.М.

Принял:

Митрохин М.А.

Пенза 2022

«УТВЕРЖДАЮ»

Зав. кафедрой ВТ

Митрохин М. А.

«___» _____ 2022 г

ЗАДАНИЕ
на курсовое проектирование по курсу

Программирование
Студенту Земсковой Виктории Михайловне Группа 21ВО2
Тема проекта: Разработка программы сложной структуры
методом нисходящего программирования

Исходные данные (технические требования) на проектирование
Наименование программы: База данных «Адресная книга».

База данных включает следующие поля:
Название, дата выхода, разработчик, издатель, наличие кооперативного режима.

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.

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 «01» октября 2022 г

Дата защиты проекта « 18 » 12 2022 г

Руководитель Митрохин Максим Александрович

Задание получил «01» октября 2022 г

Студент Земскова Виктория Михайловна

Кафедра «Вычислительная техника»

« » 2022 Г

на курсовое проектирование по курсу

методом нисходящего программирования

Наименование программы: База данных «Адресная книга».

База данных включает следующие поля:
Название, дата выхода, разработчик, издатель, наличие кооперативного режима.

- 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.

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 «01» октября 2022 г

Дата защиты проекта «___» _____ 2022 г

Руководитель Митрохин Максим Александрович

Задание получил «01» октября 2022 г

Студент Земскова Виктория Михайловна

Содержание

Введение.....	5
1 Постановка задач.....	7
2 Выбор решения.....	7
2.1 Определение необходимых модулей программ.....	7
2.2 Определение структуры файла базы данных	9
3 Описание разработки программы.....	9
4 Отладка и тестирование.....	12
5 Описание программы.....	12
6 Руководство пользователя.....	14
Заключение	18
Список используемых источников.....	19
Приложение А Листинги программы.....	20
Приложение А.1.Адресная книга.cpp.....	20
Приложение А.2. book.h.....	21
Приложение А.3. b_sort.cpp.....	22
Приложение А.4. struct.cpp.....	22
Приложение А.5. Search.cpp.....	23
Приложение А.6.Push.cpp.....	23
Приложение А.7. Pri.cpp.....	24
Приложение А.8. Pop.cpp.....	24
Приложение А.9. Load.cpp.....	25
Приложение А.10. Filtering.cpp.....	25
Приложение А.11.Download.cpp.....	26
Приложение А.12. spisok.cpp....	27
Приложение А.13. title.cpp.....	32
Приложение А.14. editing.cpp.....	33

Введение

На сегодняшний день информация – ценный источник знаний, систематизация которой, позволяет оптимизировать работу как одного человека, так и общества в целом. Компьютерные технологии во многом упрощают нашу жизнь в самых разнообразных сферах нашей деятельности. И именно они позволяют грамотно взаимодействовать с информацией, обрабатывать её и хранить на компактных носителях. Базы данных — это совокупность структур, предназначенных для хранения больших объемов информации, и программных модулей, осуществляющих управление данными, выборку, сортировку и другие действия над ними. В базах данных можно сохранять любую информацию, чтобы позже получать к ней доступ. Информация базы данных хранится в одной или нескольких таблицах.

Любая таблица с данными состоит из набора однотипных записей, расположенных друг за другом, которые можно добавлять, изменять и удалять. Каждая запись является набором именованных полей или ячеек, которые могут хранить самую разнообразную информацию (фамилия, дата рождения и пр.).

С появлением и расширением компьютерных технологий появились противоречия между ограниченными возможностями человека по восприятию и переработке информации с существующими массивами хранящейся и передаваемой информации. Для решения подобных проблем применяются автоматизированные базы данных. Они стали неотъемлемой частью практически всех компьютерных систем – от глобальной отрасли до отдельного предприятия. Для персональных компьютеров было разработано множество различных программ, предназначенных для решения познавательной потребности человека.

Данная программа является информационно-поисковой системой, позволяющей любому пользователю просматривать и добавлять информацию, а также выполнять над ней другие операции.

1 Постановка задачи

Разработать программу «База данных. Адресная книга.». Информация должна выводиться в виде таблицы со следующими полями: ФИО, год рождения, адрес проживания, домашний телефон, рабочий телефон.

Программа должна позволять пользователю осуществлять обработку данных, для этого в программе должны присутствовать следующие функции: создание новой базы данных, добавление записей, удаление записей, поиск и сортировка записей, редактирование записей, фильтрация записей по определенному критерию, сохранение текущей базы данных в файл. Меню позволяет определить пользователю возможности базы данных и осуществить дальнейшие действия. В программе должны использоваться сложные типы данных: структуры, массивы, списки, файлы. Это необходимо 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 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 2019, язык программирования Си.

2 Выбор решения

2.1. Определение необходимых модулей программы

Разработанная программы состоит из 9 модулей:

1. *struct.cpp*
2. *push.cpp*
3. *spisok.cpp*
4. *pri.cpp*
5. *title.cpp*
6. *pop.cpp*
7. *search.cpp*
8. *filtering.cpp*
9. *editing.cpp*
10. *load.cpp*
11. *download.cpp*
12. *b_sort.cpp*
13. *Адресная книга.cpp*

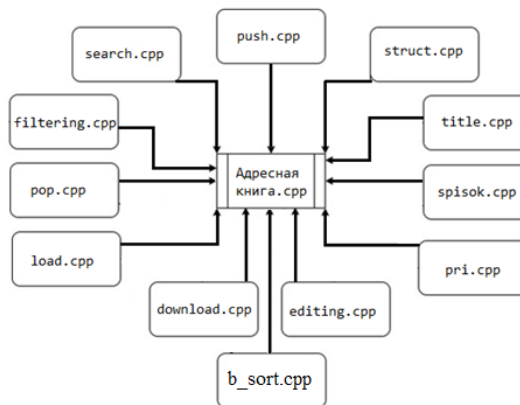


Рисунок 1 — Структура модулей программы

В файле **push.cpp** описана функция позволяющая заносить новые данные в базу данных.

В файле **struct.cpp** описана структура базы данных.

В файле title.cpp описана функция с ассемблерной вставкой, выводящая титульный лист на экран.

В файле spisok.cpp описана функция создания новой базы данных.

В файле pri.cpp описана функция, осуществляющая вывод базы данных на экран.

В файле editing.cpp описана функция редактирования базы данных.

В файле download.cpp описана функция позволяющая получить базу данных из файла.

В файле load.cpp описана функция позволяющая сохранить базу данных в файл.

В файле pop.cpp описана функция позволяющая удалить последнюю запись.

В файле filtering.cpp описана функция позволяющая осуществлять фильтрацию по выбранному полю.

В файле search.cpp описана функция, которая позволяет находить в списке запись, с помощью исходного слова, в которой имеется данное слово.

В файле b_sort.cpp описаны функции сортировки базы данных.

Файл Адресная книга.cpp является связующим звеном, реализующим вышеописанные функции. В нем строится работа всех функций, навигационного меню и его подпунктов.

2.2 Определение структуры файла базы данных

Была выбрана следующая структура:

```
typedef struct address_book
{
    char fio[32], year[32], address[32], home_phone[32], office_phone[32];
    address_book* next;
};
```

Поля fio(«ФИО»), year(«Год рождения»),address(«Адрес проживания»), home_phone(«Домашний телефон»), office_phone(«Рабочий телефон») представляют собой символьную последовательность, имеющую ограничение в 32 символа.

3 Описание разработки программы

В качестве среды программирования был выбран программный продукт Microsoft Visual Studio 2019

Интерфейс базы данных представляет собой меню. Перемещение по пунктам меню осуществляется при помощи цифр от 1 до 10, 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 '*'.

Так как при повторном выборе пункта меню «Создать новую базу данных», исходная база данных автоматически сотрется для создания новой базы данных, в навигационном меню предусмотрен пункт для обновления базы данных «Добавить запись в базу данных». Данный пункт позволяет добавить новый узел в конец существующей базы данных.

Для корректировки предусмотрена функция удаления узла, находящаяся в пункте «Удалить последнюю запись» 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 «Загрузить базу данных в файл» 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 списка на экран» позволяет в любой момент использования программы выбрать его и тем самым просмотреть содержимое существующей базы данных.

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

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

Пункт меню «Сортировка» позволяет сортировать базу данных по ФИО или году рождения.

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 `spisok.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 («ФИО», «Год рождения», «Адрес проживания», «Домашний телефон», «Рабочий телефон»). Затем были описаны функции для вывода созданной базы данных на экран – `pr()`, сохранения базы данных в файл – `load()`. Также файл `push.cpp` содержит функцию `push()`, позволяющую добавить узел в конец базы данных. Также для Функция `pop()` позволяет удалить узел динамического списка. Функция `filter()` позволяет отфильтровать базу данных по введенному фильтру по всем возможным полям. Функция `editing()` позволяет редактировать базу данных. Функция `b_sort()` позволяет сортировать базу данных по двум полям ФИО и год рождения. Функциями информации и навигационного меню является функция `main()`. Функция `main()` является главной функцией проекта и находится в файле `Адресная книга.cpp`. 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 2.

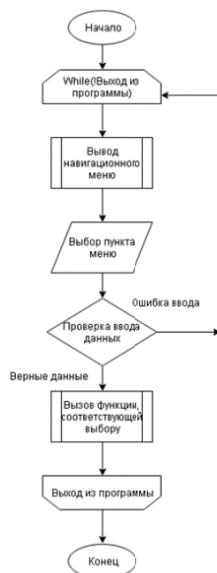


Рисунок 2— Схема программы функции main()

В зависимости от пункта навигационного меню, после выбора может вызваться либо подменю, либо выполнение определенной функции. Алгоритм описанных действий представлен ниже на рисунке 3.

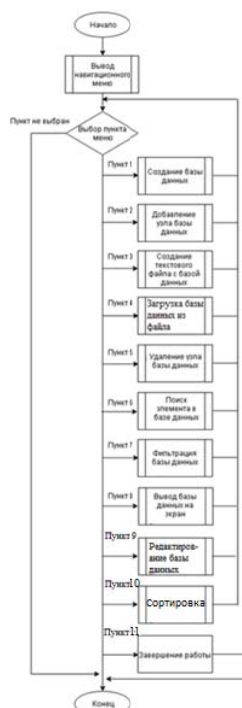


Рисунок 3— Схема навигационного меню

4 Отладка и тестирование

При создании проекта использовалась программа Microsoft Visual Studio 2019. Она имеет все необходимые для разработки и отладки программы. Для отладки программы использовались: точки останова, 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:

Основные функции для работы с базой данных:

1. Создать новую базу данных
2. Добавить запись в базу данных
3. Загрузить базу данных в файл
4. Получить базу данных из файла
5. Удалить последнюю запись
6. Поиск
7. Фильтрация
8. Вывод базы данных на экран
9. Редактирование базы данных
10. Сортировка БД
11. Выход

Ниже представлены схемы некоторых функций:



Рисунок 4— Схема функции `push.crr`

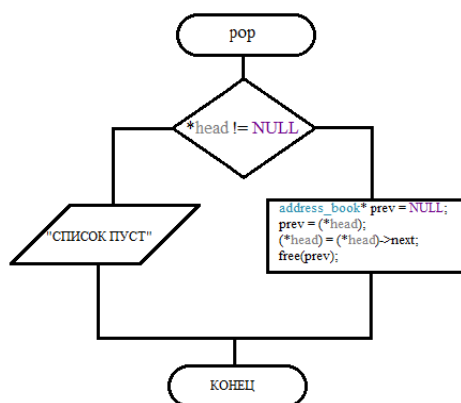


Рисунок 5— Схема функции `pop.crr`

6 Руководство пользователя

При запуске программы у нас появляется титульный лист.

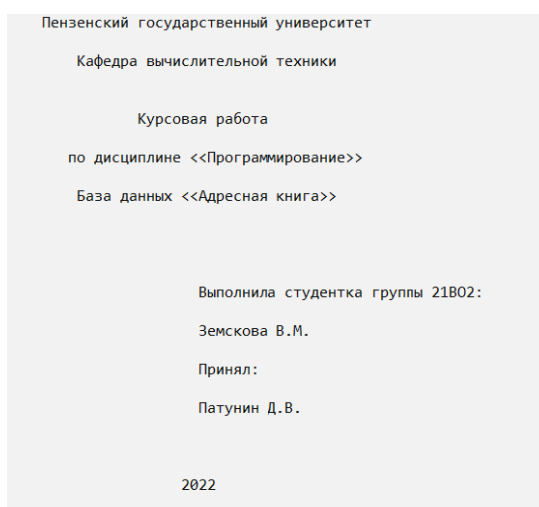


Рисунок 6— Титульный лист

Программа предназначена для хранения данных для адресной книги. Программа имеет простой и удобный интерфейс. Пользователю предоставляются все возможности для работы с базой данных, описанные в пользовательском меню.

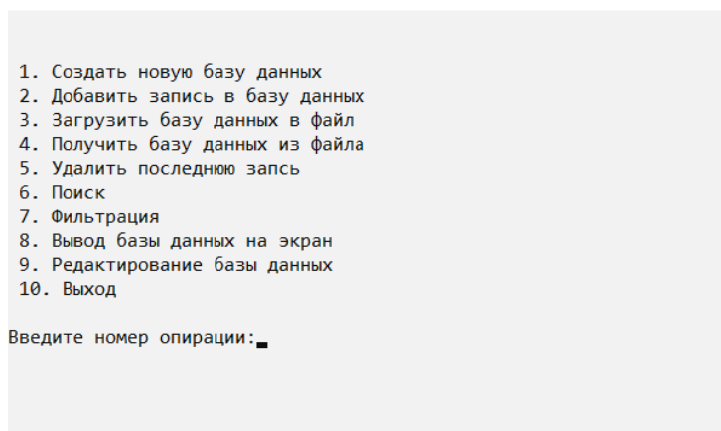


Рисунок 7— Меню программы

Чтобы создать новую базу данных будет необходимо воспользоваться пунктом 1 навигационного меню. После чего необходимо заполнить требуемые поля или выбрать поле из списка.

```
Введите ФИО:Иванов Иван Иванович
Введите год рождения:1983
Введите адрес проживания:Пенза Красная 40
Введите домашний телефон:893021737219
Введите рабочий телефон:748327932
```

Рисунок 8— Создание базы данных

Чтобы завершить создание базы данных нужно ввести звездочку.

```
Чтобы завершить создание базы данных введите *
```

Рисунок 9— Запрос на действие

Чтобы вывести на экран созданную базу данных необходимо воспользоваться 8-ым пунктом меню.

	ФИО	Год рождения	Адрес проживания	Домашний телефон	Рабочий телефон
1	Сёмин Андрей Викторович	1989	Москва Мира 156	8392018	4832098432
2	Иванов Иван Иванович	1978	Пенза Красная 40	839271432	37219832
Для продолжения нажмите любую клавишу . . .					

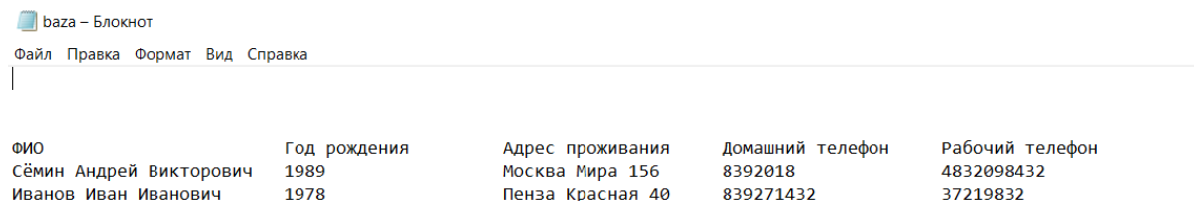
Рисунок 10— Вывод базы данных

Для создания файла и записи в него созданной базы данных предназначен пункт меню 3

Введите имя файла для записи базы данных:data.txt

Рисунок 11— Запись базы данных в файл

Результат будет записан в файле data.txt.

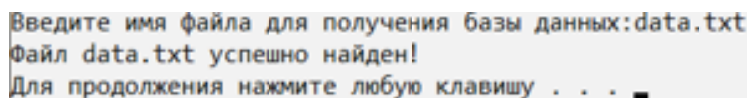


Файл Правка Формат Вид Справка

ФИО	Год рождения	Адрес проживания	Домашний телефон	Рабочий телефон
Сёмин Андрей Викторович	1989	Москва Мира 156	8392018	4832098432
Иванов Иван Иванович	1978	Пенза Красная 40	839271432	37219832

Рисунок 12— Запись базы данных в файл data.txt

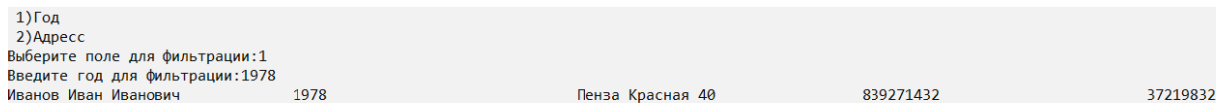
Для получения базы данных из файла нужно воспользоваться пунктом 4 и ввести название файла, из которого будет получена база данных.



```
Введите имя файла для получения базы данных:data.txt
Файл data.txt успешно найден!
Для продолжения нажмите любую клавишу . . .
```

Рисунок 13— 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.



1)Год
2)Адресс
Выберите поле для фильтрации:1
Введите год для фильтрации:1978

Иванов Иван Иванович	1978	Пенза Красная 40	839271432	37219832
----------------------	------	------------------	-----------	----------

Рисунок 14— Фильтрация

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

Если узел необходимо добавить, то следует воспользоваться 2-ым пунктом меню. Этот пункт позволяет добавлять узел в конец базы данных.

Введите ФИО:Иванов Иван Иванович
Введите год рождения:1983
Введите адрес проживания:Пенза Красная 40
Введите домашний телефон:893021737219
Введите рабочий телефон:748327932

Рисунок 15— Добавление узла базы данных

Пункт 6 в главное меню осуществляет поиск по слову.
Пользователь должен ввести слово, после чего выведется результат.

Введите слово для поиска:Иван
Результат поиска:Иванов Иван Иванович 1978 Пенза Красная 40 839271432 37219832
Для продолжения нажмите любую клавишу . . .

Рисунок 16— Поиск по слову

Пункт 9 в главном меню позволяет редактировать записи в базе данных.

	ФИО	Год рождения	Адрес проживания	Домашний телефон	Рабочий телефон
1	Иванов Иван Иванович	1978	1578	1978	1978
2	Сёмин Андрей Викторович	1989	Москва Мира 156	8392018	4832098432

Выберите строку для редактирования:
1
Выберите поле для редактирования:
1)ФИО
2)Год рождения
3)Адрес проживания
4)Домашний телефон
5)Рабочий телефон
1
Введите слово для замены:Сидоров Иван Иванович

Рисунок 17— Редактирование базы данных

Последний пункт меню под номером 10 выполняет сортировку по ФИО или по Году рождения.

№	ФИО	Год рождения	Адрес проживания	Домашний телефон	Рабочий телефон
1	Петракова Елизавета	2001	Мира 76	9489374329	74394320
2	Кузьмичёва Полина	2004	Куйбышева 40	3843905	859043
3	Игонин Артём	2001	Красная 5	89473284320	7937219321

Для продолжения нажмите любую клавишу . . .

Рисунок 18— Список до сортировки

Выбрали 1 пункт и отсортировали список по ФИО.

1) Сортировка по ФИО
2) Сортировка по году рождения
Выберите сортировку:1

№	ФИО	Год рождения	Адрес проживания	Домашний телефон	Рабочий телефон
1	Игонин Артём	2001	Красная 5	89473284320	7937219321
2	Кузьмичёва Полина	2004	Куйбышева 40	3843905	859043
3	Петракова Елизавета	2001	Мира 76	9489374329	74394320

Для продолжения нажмите любую клавишу . . .

Рисунок 19— Результат сортировки по ФИО

Выбрали 2 пункт и отсортировали по году рождения.

1) Сортировка по ФИО					
2) Сортировка по году рождения					
Выберите сортировку:2					
№	ФИО	Год рождения	Адрес проживания	Домашний телефон	Рабочий телефон
1	Игонин Артём	2001	Красная 5	89473284320	7937219321
2	Петракова Елизавета	2001	Мира 76	9489374329	74394320
3	Кузьмичёва Полина	2004	Куйбышева 40	3843905	859043
Для продолжения нажмите любую клавишу . . .					

Рисунок 20— Результат сортировки по году рождения

Последний 11 пункт осуществляет выход из программы.

Заключение

При выполнении данной курсовой работы были получены навыки работы с многомодульными программами, сложными типами данных, файлами, механизмами разработки меню и осуществления связи между каждым его пунктом и функциями для работы с базами данных. Получены и «усовершенствованы» навыки отладки тестирования программы в среде программирования Microsoft Visual Studio 2019.

В процессе выполнения курсовой работы была создана программа «База данных. Адресная книга», предназначенная для хранения, передачи и обработки данных о ФИО, годе рождения, адресе проживания, домашнем и рабочем телефонах людей.

Список используемых источников

1. Курс программирования на языке СИ. С. Фомин, В. Подбельский
2. <http://all-ht.ru>
3. <https://ru.stackoverflow.com>
4. <https://metanit.com>
5. Программирование на С в примерах и задачах. А. Васильев.

Приложение А

Листинг программы:

Приложение А.1.Адреснаякнига.cpp

```
#include"stdafx.h"

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

voidmain()
{
    system("color f0");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    title();
    system("pause");
    int count;
    address_book* head = NULL;
    for(;;)
    {
        system("cls");
        printf("\n\n 1. Создать новую базу данных\n 2. Добавить запись в базу
данных\n 3. Загрузить базу данных в файл\n 4. Получить базу данных из файла\n 5. Удалить
последнюю запись\n 6. Поиск\n 7. Фильтрация\n 8. Вывод базы данных на экран\n 9.
Редактирование базы данных\n 10. Сортировка\n 11. Выход\n\n");
        printf("Введите номер опирации:");
        scanf("%d", &count);
        switch (count)
        {
            case 1:
                system("cls");
                spisok(&head);
```

```

        break;
    case 2:
    {
        system("cls");
        char fio[32],    year[32],    address[32],    home_phone[32],
office_phone[32];
        printf("Введите год рождения:");
        gets(year);
        printf("Введите ФИО:");
        gets(fio);
        printf("Введите адрес проживания:");
        gets(address);
        printf("Введите домашний телефон:");
        gets(home_phone);
        printf("Введите рабочий телефон:");
        gets(office_phone);
        push(&head, fio, year, address, home_phone, office_phone);
        break;
    }
    case 3:
        system("cls");
        load(&head);
        break;
    case 4:
        system("cls");
        download(&head);
        system("pause");
        break;
    case 5:
        system("cls");
        pop(&head);
        system("pause");
        break;
    case 6:
        system("cls");
        search(&head);

```

```

        system("pause");
        break;
    case 7:
        system("cls");
        filtering(&head);
        system("pause");
        break;
    case 8:
        system("cls");
        fflush(stdin);
        pri(&head);
        system("pause");
        break;
    case 9:
        system("cls");
        editing(&head);
        system("pause");
        break;
    case 10:
        system("cls");
        sort(&head);
        pri(&head);
        break;
    case 11:
        exit(0);
        break;
    }
}
}

```

ПриложениеА.2. book.h

```

#include"struct.cpp"
#include"push.cpp"
#include"spisok.cpp"
#include"pri.cpp"
#include"title.cpp"

```

```

#include"pop.cpp"
#include"search.cpp"
#include"filtering.cpp"
#include"editing.cpp"
#include"load.cpp"
#include"download.cpp"
#include"b_sort.cpp"

voidsort(address_book** head);
voidsort_fio(address_book** head);
voidsort_year(address_book** head);
voidspisok(address_book** head);
voiddownload(address_book** head);
voidtitle();
voidpush(address_book** head, char* fio, char* year, char* address, char* home_phone,
char* office_phone);
voidpop(address_book** head);
voidsearch(address_book** head);
voidfiltering(address_book** head);
voidediting(address_book** head);
voidload(address_book** head);
voiddownload(address_book** head);
voidpri(address_book** head);

```

Приложение А.3. b_sort.cpp

```

voidsort_year(address_book** head)
{
    charfio[32], year[32], address[32], home_phone[32], office_phone[32];
    int size = 0;
    address_book* ptr = (*head);
    while (ptr != NULL)
    {
        ptr = ptr->next;
        size++;
    }
    ptr = (*head);

```

```

for (int j = 1; j < size; j++)
{
    ptr = (*head);
    for (inti = 0; i < size - j; i++)
    {
        if (atoi(ptr->year) > atoi(ptr->next->year))
        {
            strcpy(fio, ptr->fio);
            strcpy(year, ptr->year);
            strcpy(address, ptr->address);
            strcpy(home_phone, ptr->home_phone);
            strcpy(office_phone, ptr->office_phone);

            strcpy(ptr->fio, ptr->next->fio);
            strcpy(ptr->year, ptr->next->year);
            strcpy(ptr->address, ptr->next->address);
            strcpy(ptr->home_phone, ptr->next->home_phone);
            strcpy(ptr->office_phone, ptr->next->office_phone);

            strcpy(ptr->next->fio, fio);
            strcpy(ptr->next->year, year);
            strcpy(ptr->next->address, address);
            strcpy(ptr->next->home_phone, home_phone);
            strcpy(ptr->next->office_phone, office_phone);
        }
        ptr = ptr->next;
    }
}

```

```

void sort_fio(address_book** head)
{
    char fio[32], year[32], address[32], home_phone[32], office_phone[32];
    int size = 0;
    address_book* ptr = (*head);
    while (ptr != NULL)

```

```

    {
        ptr = ptr->next;
        size++;
    }
    ptr = (*head);
    for (int j = 1; j < size; j++)
    {
        ptr = (*head);
        for (inti = 0; i< size - j; i++)
        {
            if (ptr->fio[0] > ptr->next->fio[0])
            {
                strcpy(fio, ptr->fio);
                strcpy(year, ptr->year);
                strcpy(address, ptr->address);
                strcpy(home_phone, ptr->home_phone);
                strcpy(office_phone, ptr->office_phone);

                strcpy(ptr->fio, ptr->next->fio);
                strcpy(ptr->year, ptr->next->year);
                strcpy(ptr->address, ptr->next->address);
                strcpy(ptr->home_phone, ptr->next->home_phone);
                strcpy(ptr->office_phone, ptr->next->office_phone);

                strcpy(ptr->next->fio, fio);
                strcpy(ptr->next->year, year);
                strcpy(ptr->next->address, address);
                strcpy(ptr->next->home_phone, home_phone);
                strcpy(ptr->next->office_phone, office_phone);
            }
            ptr = ptr->next;
        }
    }
}

```

```

void sort(address_book** head)

```

```

{
    int a;
    printf("\n\n1)    Сортировка    по    ФИО\n2)    Сортировка    по    году
рождения\n\nВыберите сортировку:");
    scanf("%d", &a);
    switch (a)
    {
    case 1:
        sort_fio(head);
        pri(head);
        system("pause");
        break;
    case 2:
        sort_year(head);
        pri(head);
        system("pause");
        break;
    default:
        break;
    }
}

```

ПриложениеА.4. struct.cpp

```

typedef struct address_book
{
    char fio[32], year[32], address[32], home_phone[32], office_phone[32];
    address_book* next;
};

```

ПриложениеА.5. Search.cpp

```

void search(address_book** head)
{
    char name[32];
    address_book* ptr = (*head);
    fflush(stdin);
    printf("Введите слово для поиска:");
}

```

```

    gets(name);
    if (ptr == NULL)
        printf("СПИСОК ПУСТ!");
    else
        printf("Результат поиска:");
    while (ptr != NULL)
    {
        if (strstr(ptr->fio, name) != NULL)
            printf("%-32s %-32s %-32s %-32s %-32s\n", ptr->fio, ptr->year, ptr->address, ptr->home_phone, ptr->office_phone);
        if (strstr(ptr->year, name) != NULL)
            printf("%-32s %-32s %-32s %-32s %-32s\n", ptr->fio, ptr->year, ptr->address, ptr->home_phone, ptr->office_phone);
        if (strstr(ptr->address, name) != NULL)
            printf("%-32s %-32s %-32s %-32s %-32s\n", ptr->fio, ptr->year, ptr->address, ptr->home_phone, ptr->office_phone);
        if (strstr(ptr->home_phone, name) != NULL)
            printf("%-32s %-32s %-32s %-32s %-32s\n", ptr->fio, ptr->year, ptr->address, ptr->home_phone, ptr->office_phone);
        if (strstr(ptr->office_phone, name) != NULL)
            printf("%-32s %-32s %-32s %-32s %-32s\n", ptr->fio, ptr->year, ptr->address, ptr->home_phone, ptr->office_phone);
        ptr = ptr->next;
    }
}

```

Приложение А.6. Push.cpp

```

void push(address_book** head, char* fio, char* year, char* address, char* home_phone, char* office_phone)
{
    address_book* tmp;
    tmp = new address_book;
    strcpy(tmp->fio, fio);
    strcpy(tmp->year, year);
    strcpy(tmp->address, address);

```

```

strcpy(tmp->home_phone, home_phone);
strcpy(tmp->office_phone, office_phone);
tmp->next = (*head);
(*head) = tmp;
}

```

Приложение A.7. Pri.cpp

```

void pri(address_book** head)
{
    if (*head != NULL)
    {
        int i = 1;
        address_book* ptr = (*head);
        ptr = (*head);
        printf("    %-32s %-32s %-32s %-32s %-32s\n", "ФИО", "Год рождения",
"Адрес проживания", "Домашний телефон", "Рабочий телефон");
        while (ptr != NULL)
        {
            printf("%-5d %-32s %-32s %-32s %-32s %-32s\n", i, ptr->fio, ptr-
>year, ptr->address, ptr->home_phone, ptr->office_phone);
            ptr = ptr->next;
            i++;
        }
    }
    else
    {
        printf("СПИСОК ПУСТ!");
    }
}

```

Приложение A.8. Pop.cpp

```

void pop(address_book** head)
{
    if ((*head) != NULL)
    {
        address_book* prev = NULL;

```

```

        prev = (*head);
        (*head) = (*head)->next;
        free(prev);
    }
    else
    {
        printf("\nСПИСОК ПУСТ\n\n");
    }
}

```

Приложение A.9. Load.cpp

```

void load(address_book** head)
{
    FILE* S;
    address_book* ptr = (*head);
    char name[32];
    fflush(stdin);
    fprintf(stdout, "%s", "Введите имя файла для записи базы данных:");
    scanf("%s", &name);
    while ((S = fopen(name, "w")) == (NULL))
    {
        fprintf(stdout, "%s", "Указанный файл не найден!\nВведите имя файла  
для записи базы данных еще раз:");
        scanf("%s", &name);
    }
    fprintf(stdout, "Файл %s успешно найден!\n", name);
    if ((*head) != NULL)
    {
        address_book* ptr = *head;
        fprintf(S, "\n\n\n%-25s  %-20s  %-20s  %-20s  %-20s\n", "ФИО", "Год  
рождения", "Адрес проживания", "Домашний телефон", "Рабочий телефон");
        while (ptr != NULL)
        {
            fprintf(S, "%-25s  %-20s  %-20s  %-20s  %-20s\n", ptr->fio, ptr->year,
ptr->address, ptr->home_phone, ptr->office_phone);
            ptr = ptr->next;
        }
    }
}

```

```

    }
}
else
{
    printf("\nСПИСОК ПУСТ\n\n");
}
fclose(S);
}

```

Приложение А.10. Filtering.cpp

```

void filtering(address_book** head)
{
    address_book* ptr = (*head);
    fflush(stdin);
    fflush(stdout);
    char number[32];
    int i;
    printf(" 1)Год\n 2)Адресс\n Выберите поле для фильтрации:");
    scanf("%d", &i);
    fflush(stdin);
    switch (i)
    {
    case 1:
        fflush(stdin);
        printf("Введите год для фильтрации:");
        gets(number);
        while (ptr != NULL)
        {
            if (strstr(ptr->year, number) != NULL)
                printf("%-32s %-32s %-32s %-32s %-32s\n", ptr->fio, ptr->year, ptr->address, ptr->home_phone, ptr->office_phone);
            ptr = ptr->next;
        }
        break;

    case 2:

```

```

fflush(stdin);
printf("Введите улицу, дом или город для фильтрации:");
gets(number);
while (ptr != NULL)
{
    if (strstr(ptr->address, number) != NULL)
        printf("%-32s %-32s %-32s %-32s %-32s\n", ptr->fio, ptr-
>year, ptr->address, ptr->home_phone, ptr->office_phone);
    ptr = ptr->next;
}
break;
default: printf("Неправильный ввод");
}
}

```

Приложение A.11. Download.cpp

```

void download(address_book** head)
{
    FILE* S;
    char name[32], fio[32], year[32], address[32], home_phone[32], office_phone[32];
    *head = NULL;
    fprintf(stdout, "%s", "Введите имя файла для получения базы данных:");
    scanf("%s", &name);
    while ((S = fopen(name, "r")) == (NULL))
    {
        fprintf(stdout, "%s", "Указанный файл не найден!\nВведите имя файла
для получения базы данных еще раз:");
        scanf("%s", &name);
    }
    fprintf(stdout, "Файл %s успешно найден!\n", name);
    if (!feof(S))
    {
        fseek(S, 117, SEEK_SET);
        while (!feof(S))

```

```

        {
            fgets(fio, 26, S);
            fseek(S, 1, SEEK_CUR);
            fgets(year, 21, S);
            fseek(S, 1, SEEK_CUR);
            fgets(address, 21, S);
            fseek(S, 1, SEEK_CUR);
            fgets(home_phone, 21, S);
            fseek(S, 1, SEEK_CUR);
            fgets(office_phone, 22, S);
            if(!feof(S))
                push(head, fio, year, address, home_phone, office_phone);
        }
    }
else
{
    printf("\nСПИСОК ПУСТ\n\n");
}
fclose(S);
}

```

Приложение А.12. spisok.cpp

```

void spisok(address_book** head)
{
    fflush(stdin);
    char fio[32], year[32], address[32], home_phone[32], office_phone[32];
    printf("Введите ФИО:");
    gets(fio);
    for(;;)
    {
        fflush(stdin);
        printf("Введите год рождения:");
        gets(year);
        printf("Введите адрес проживания:");
        gets(address);
    }
}

```

```

        printf("Введите домашний телефон:");
        gets(home_phone);
        printf("Введите рабочий телефон:");
        gets(office_phone);
        push(head, fio, year, address, home_phone, office_phone);
        printf("Чтобы завершить создание базы данных введите *\nВведите
ФИО:");

        gets(fio);
        if (fio[0] == '*')
            break;
    }
}

```

Приложение А.13. title.cpp

```

void title()
{
    printf("
Пензенский государственный университет\n\n
Кафедра вычислительной техники\n\n
Курсовая работа\n\n
под дисциплине <<Программирование>>\n\n
База данных <<Адресная книга>>\n\n\n\n
Выполнила студентка группы 21ВО2:\n\n
Земскова В.М.\n\n
Принял:\n\n
Патунин Д.В.\n\n\n\n
2022\n\n");
    _getch();
}

```

Приложение А.14. editing.cpp

```

void editing(address_book** head)
{
    address_book* ptr = (*head);
    int number, i = 0;
    char name[32];
    pri(head);
    if (ptr != NULL)
    {
        while (ptr != NULL)
        {

```

```

        i++;
        ptr = ptr->next;
    }
    i--;
    ptr = (*head);
    printf("Выберите строку для редактирования:");
    scanf("%d", &number);
    i -= number;
    for (; i > 0; i--)
        ptr = ptr->next;
    printf("Выберите поле для редактирования:\n 1)ФИО\n 2)Год
рождения\n 3)Адрес проживания\n 4)Домашний телефон\n 5)Рабочий телефон\n");
    scanf("%d", &i);
    fflush(stdin);
    printf("Введите слово для замены:");
    gets(name);
    switch (i)
    {
    case 1:
        strcpy(ptr->fio, name);
    case 2:
        strcpy(ptr->year, name);
    case 3:
        strcpy(ptr->address, name);
    case 4:
        strcpy(ptr->home_phone, name);
    case 5:
        strcpy(ptr->office_phone, name);
    default: break;
    }
}
else {
    printf("СПИСОК ПУСТ!");
}
}

```