

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

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту
по курсу «Программирование»
на тему «Разработка программы сложной структуры
методом нисходящего программирования»

Выполнил
студент группы 21ВО1
Антонов Д.А

Принял
Митрохин М. А.

Пенза 2022

Оглавление

Введение	5
1 Постановка задачи	6
1.1 Требования к программе	6
2 Выбор решения	7
2.2 Определение структуры файла базы данных	8
2.3 Описание разработки программы	8
3 Отладка и тестирование	15
4 Описание программы	16
4.1 Разработка приложения	16
4.2 Разработка меню	17
5 Руководство пользователя	18
Заключение	22
Список использованных источников	23
Приложение А Исходный код программы	24
Приложение Б Содержимое файла	28

Введение

Сегодня трудно себе представить сколько-нибудь значимую информационную систему, которая не имела бы в качестве основы или важной составляющей базу данных. Концепции и технологии баз данных складывались постепенно и всегда были тесно связаны с развитием систем автоматизированной обработки информации. Создание баз данных после появления реляционного подхода превратилось из искусства в науку, но, как показала практика последних лет, все же окончательно его не исключившая. Тем не менее, сейчас это вполне сложившаяся дисциплина (хотя являющаяся скорее инженерной, чем чисто научной), основанная на достаточно формализованных подходах и включающая широкий спектр приемов и методов создания баз данных.

База данных (далее БД) — это поименованная совокупность взаимосвязанных данных, управляемых специальной системой, называемой СУБД. СУБД представляет собой совокупность специальных языковых и программных средств, облегчающих пользователям выполнение всех операций, связанных с организацией хранения данных, их корректировкой и доступом к ним. СУБД служит, по существу, посредником между пользователем и БД.

Задачей данной курсовой работы стало проектирование базы данных «Список книг». При проектировании базы данных будут получены навыки написания программ, содержащие сложные структуры — базы данных. Программа может использоваться людьми, не имеющими квалификации программиста.

1 Постановка задачи

Необходимо разработать программу – базу данных «Список книг». База данных включает в себя следующие поля: Название, жанр, автор, издатель, год выпуска.

Программа должна позволять создавать новую базу данных, и загружать ранее сохраненную. После создания или загрузки базы данных программа должна иметь возможность добавления/удаления записи, осуществлять поиск записи, также должен быть доступен просмотр всех записей. Для реализации необходимо продумать интерфейс, интуитивно понятный для пользователя и определить названия пунктов меню. Также важно продумать структуру работы программа и механизмы взаимодействия с функциями.

1.1 Требования к программе

В программе используются сложные типы данных - структуры и файлы. Программа работает в текстовом режиме и использует устройства ввода информации – клавиатуру и мышь. Пользовательский интерфейс разработан на основе меню. Программа исправно функционирует на операционной системе Windows. Программа была разработана на базе среды Microsoft Visual Studio 2019 и написана на языке программирования Си.

2 Выбор решения

Разработанная программа состоит из трех модулей. Схема взаимодействия модулей приведена на рисунке 1.



Рисунок 1 – структура программы

Модуль «Вывод главного меню» содержит в себе функцию, которая выводит главное меню на экран.

Модуль «Вывод доп. меню» содержит в себе функцию, которая выводит главное дополнительное меню на экран.

Модуль «Обработка» содержит в себе функцию, которая позволяет обрабатывать введенное значение и взаимодействовать с меню.

Модуль «Сохранение БД в файл» содержит в себе функцию, которая позволяет сохранять БД в файл.

Модуль «Добавление записи» содержит в себе функцию, которая позволяет добавлять запись в БД.

Модуль «Удаление записи» содержит в себе функцию, которая позволяет удалять запись из базы данных.

Модуль «Поиск записи» содержит в себе функцию, которая позволяет найти запись в базе данных.

Модуль «Отображение записи» содержит в себе функцию, которая позволяет просматривать записи в базе данных.

2.2 Определение структуры файла базы данных

Для структурирования данных внутри файла была выбрана следующая структура данных:

```
typedef struct list_  
{  
    int data;  
    char name[100];  
    char genre[100];  
    char author[100];  
    char izd[100];  
    char year[100];  
    struct list_* next;  
} list;
```

Программа умеет делать следующее:

- 1) Создавать базу данных;
- 2) Сохранять базу данных;
- 3) Добавлять запись в базу данных;
- 4) Удалять запись из базы данных;
- 5) Находить запись в базе данных;
- 6) Выводить на экран базу данных.

2.3 Описание разработки программы

Для написания данной программы будет использован язык программирования Си. Данный язык является распространенным языком программирования. Язык Си в настоящее время является стандартным базовым языком, с которого начинают свое знакомство с программированием студенты первых курсов вузов.

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

При запуске программы на экран выводится заставка-приветствие и первоначальное меню. Пользователь может выбрать пункты «Создать новую базу данных» или вовсе выйти из программы посредством пункта «Выход». Для взаимодействия с меню пользователь должен ввести соответствующую цифру, которая привязана к одному из пунктов.

Когда пользователь выбрал одно из двух пунктов первоначального меню, на экран выводится вторичное меню, которое позволяет работать с базой

данных. Пользователь может добавить, удалить, найти, отобразить записи с помощью пунктов «Добавить книгу», «Удалить книгу», «Поиск книги», «Отобразить все книги» соответственно. Также пользователь может сохранить базу данных в файл с помощью пункта «Сохранить базу данных в файл». Пользователь может перейти обратно в первоначальное меню с помощью пункта «Назад».

Первоначально нужно написать основное и дополнительное меню, а далее реализовывать функции, которые будут выполнять действия при взаимодействии с меню.

Первоначальное меню вызывается посредством функции «printMainMenu». Она отвечает за вывод первоначального меню и заставки. Блок-схема вывода меню приведена на рисунке 2.



Рисунок 2 – Вывод первоначального меню

Дополнительное меню вызывается посредством функции «printMenu». Она отвечает за вывод дополнительного меню. Блок-схема вывода дополнительного меню приведена на рисунке 3.



Рисунок 3 – Вывод дополнительного меню

Для обработки выбора пунктов меню была реализована функция `getVariant`, которая принимает значение и обрабатывает его. Блок-схема функции `getVariant` приведена на рисунке 3.

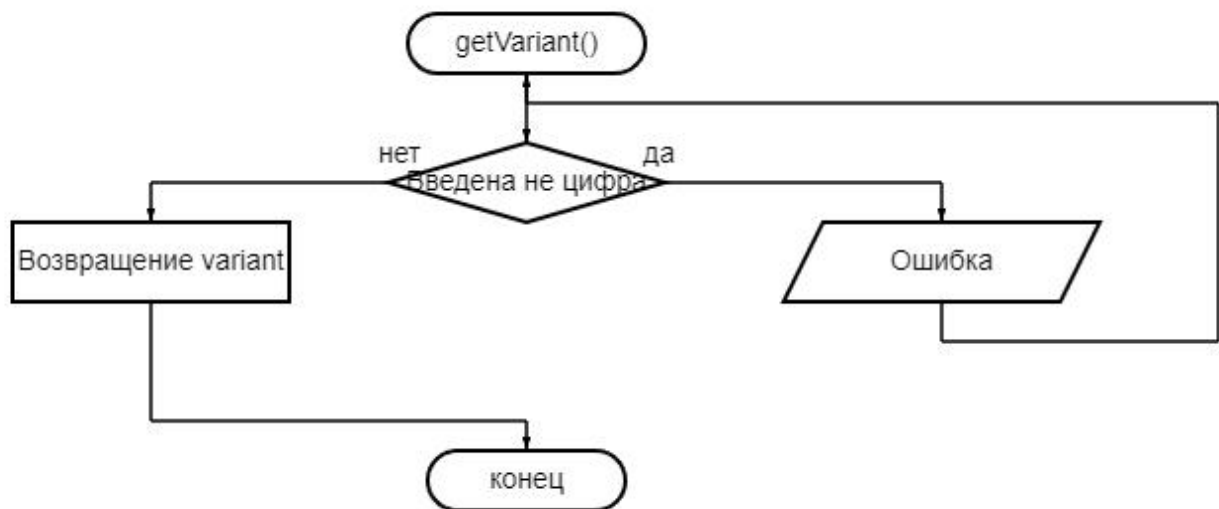


Рисунок 3 – Схема функции `getVariant`

Для обработки действий пользователя при работе с меню была добавлена переменная `variant`, которая в зависимости от введенного значения вызывала нужные функции.

Для создания базы данных была реализована функция `list_create`, блок-схема которой приведена на рисунке 4.

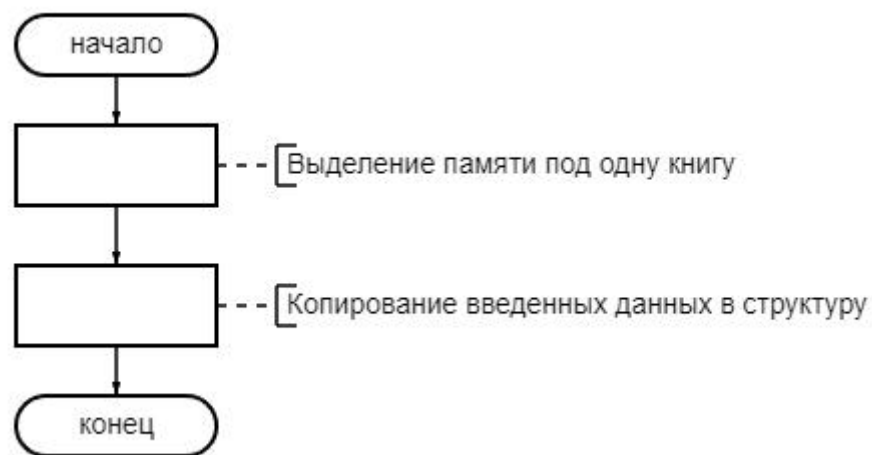
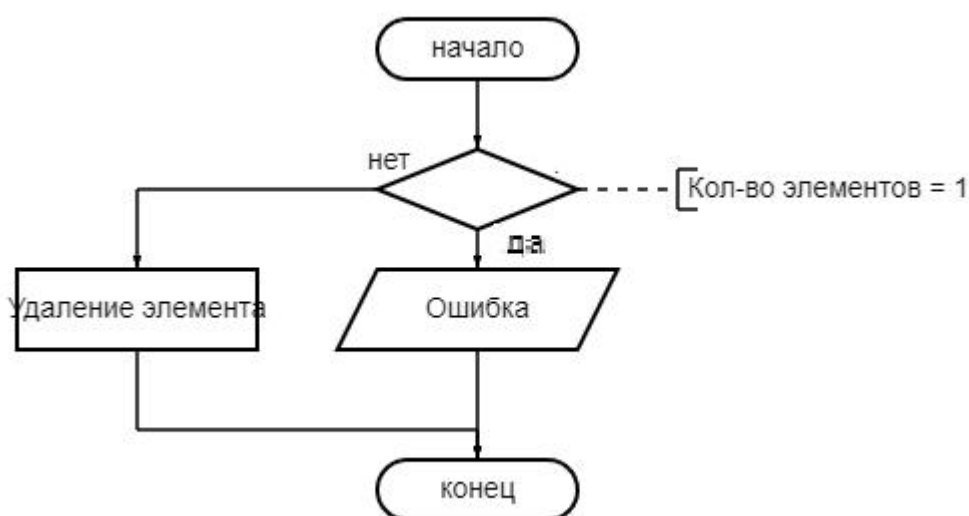


Рисунок 4

Для удаления книги из базы данных была реализована функция `list_delete`, блок-схема которой приведена на рисунке 6.



Рисуно

к 6 – удаление книги из базы данных

Для поиска книги в базе данных была реализована функция `list_search`, блок-схема которой приведена на рисунке 7.

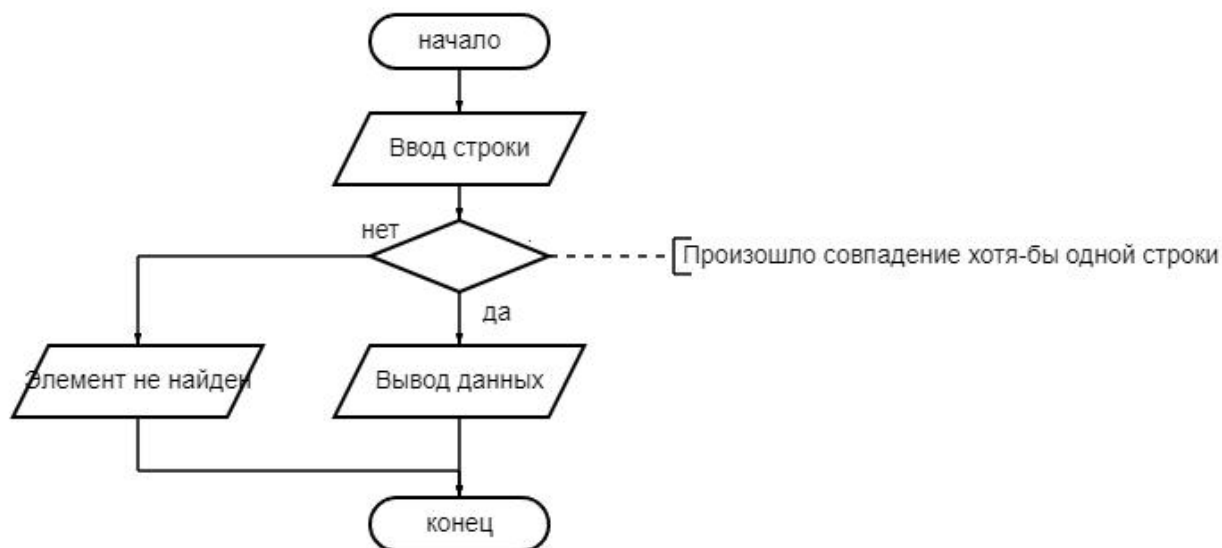


Рисунок 7 – поиск книги в базе данных

Для вывода книг в базе данных была реализована функция `list_print`, блок-схема которой приведена на рисунке 8.



Рисунок 8 – вывод книг из базы данных

Для сохранения базы данных в файл была реализована функция `saveToFile`, блок-схема которой приведена на рисунке 9.

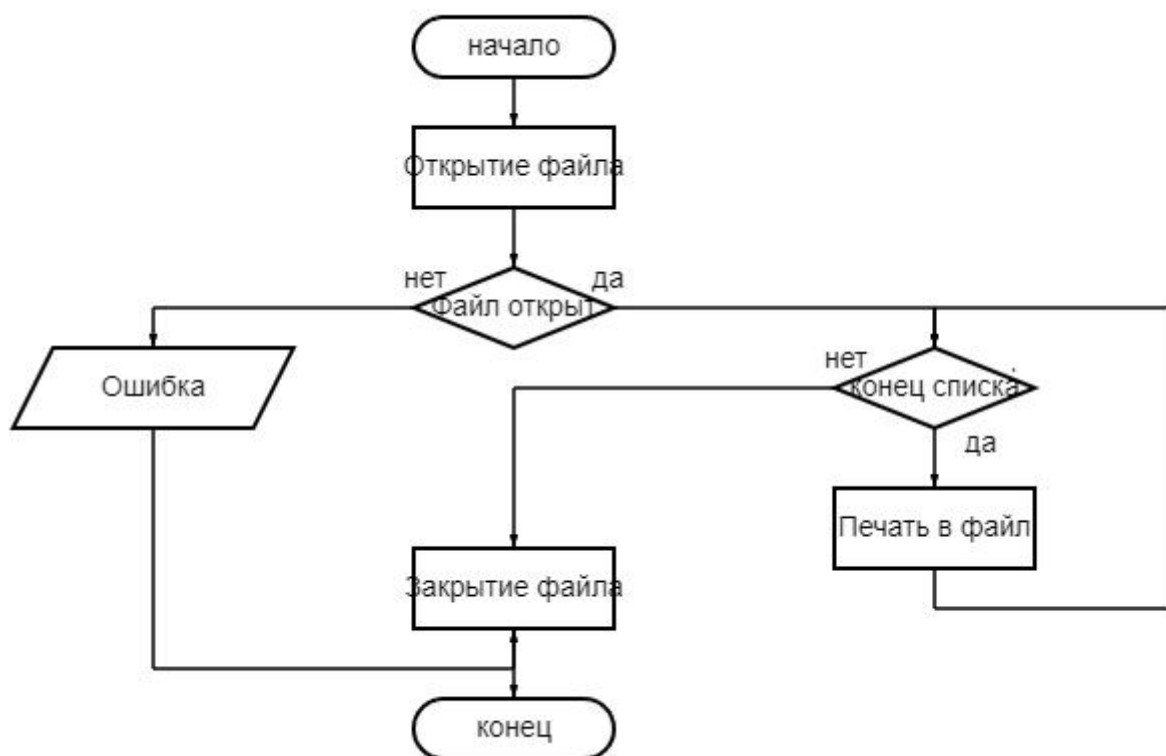


Рисунок 9 – Сохранение базы данных в файл

Блок-схема работы главной программы приведена на рисунке 10.

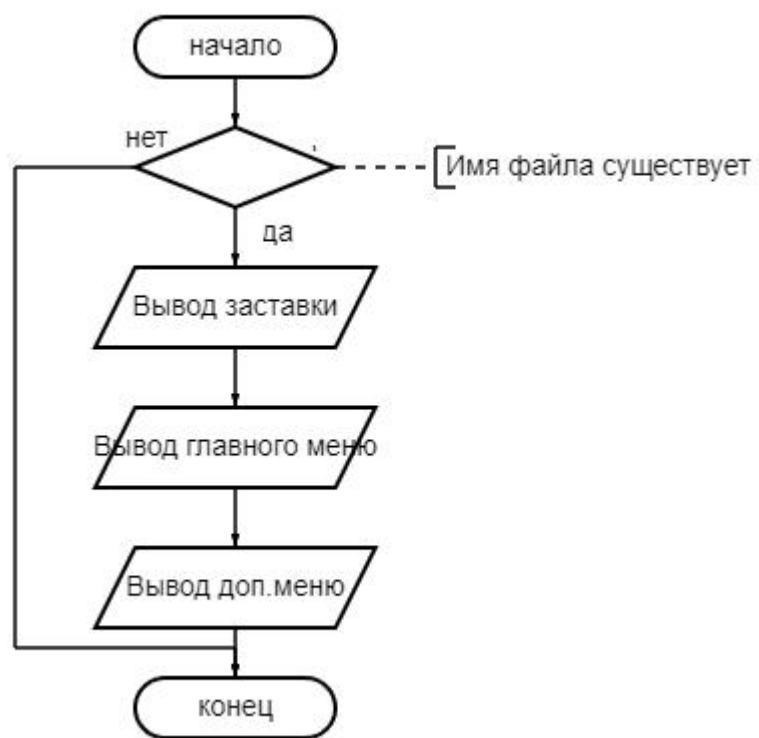


Рисунок 10 – Схема главной программы

4 Описание программы

4.1 Разработка приложения

При запуске программы на экран выводится заставка-приветствие и начальное меню, к этому моменту программа готова к выполнению своих функций.

Описание состояний программы (первоначальное меню):

Таблица 1 – описание состояний программы в основном меню

Клавиша	Действие пользователя	Действие программы
1	Выбран пункт «Создать БД»	Программа вызывает функцию printMainMenu() и list_create(list*,name,genre,author,izd,year)
2	Выбран пункт «Выход»	Программа закрывается

Описание состояний программы (дополнительное меню):

Таблица 2 – описание состояний программы в дополнительном меню

а

Клавиша	Действие пользователя	Действие программы
1	Выбран пункт «Добавить книгу»	Программа вызывает функцию list_add_begin(list* head, name,genre,author,izd,year)
2	Выбран пункт «Удалить книгу»	Программа вызывает функцию list_delete(list *head)
3	Выбран пункт «Поиск книги»	Программа вызывает функцию list_search(list *head)
4	Выбран пункт «Отобразить все книги»	Программа вызывает функцию list_print(list *head)
5	Выбран пункт «Сохранить БД в файл»	Программа вызывает функцию saveToFile(list *head)
6	Выбран пункт «Назад»	Программа вызывает функцию printMainMenu()

4.2 Разработка меню

Основное и дополнительное меню реализовано в функциях `printMainMenu()` и `printMenu()` соответственно.

Основное меню включает в себя 2 пункта:

- 1) Создать новую базу данных
- 2) Выход

Дополнительное меню включает в себя 7 пунктов:

- 1) Добавить книгу
- 2) Удалить книгу
- 3) Поиск книги
- 4) Отобразить все книги
- 5) Сохранить базу данных в файл
- 6) Назад

Пользователь вводит цифру, указанную в меню, и программа вызывает нужную функцию.

5 Руководство пользователя

Программа предназначена для хранения информации о различных книгах. В таблице имеются такие поля как название, жанр, автор, издатель, год издания. Программа может создавать базу данных, добавлять записи в нее и удалять их, осуществлять поиск, и записывать результат в файл.

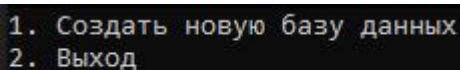
При запуске программы на экран выводится заставка. (рисунок 13)



```
База данных 'Список книг' разработана студентом Антоновым Д.А
```

Рисунок 13 – Заставка

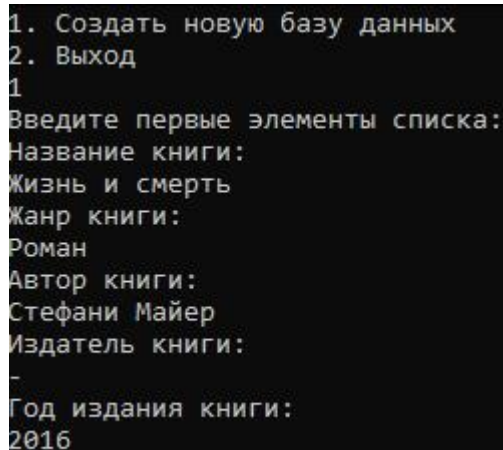
Далее выводится главное меню (рисунок 14)



```
1. Создать новую базу данных
2. Выход
```

Рисунок 14– Главное меню

Для создания базы данных необходимо выбрать пункт 1: «Создать новую базу данных» после чего вводятся данные о книге (рисунок 15).



```
1. Создать новую базу данных
2. Выход
1
Введите первые элементы списка:
Название книги:
Жизнь и смерть
Жанр книги:
Роман
Автор книги:
Стефани Майер
Издатель книги:
-
Год издания книги:
2016
```

Рисунок 15– выбран пункт «1»

Далее выводится дополнительное меню (рисунок 16)

```
1. Добавить книгу
2. Удалить книгу
3. Поиск книги
4. Отобразить все книги
5. Сохранить базу данных в файл
6. Назад
>
```

Рисунок 16 – Дополнительное меню

При выборе первого пункта в дополнительном меню пользователь вводит данные о книге (рисунок 17)

```
CS\ D:\Хранение данных\ПГУ\2 курс\Программ
1. Добавить книгу
2. Удалить книгу
3. Поиск книги
4. Отобразить все книги
5. Сохранить базу данных в файл
6. Назад
>1
Название книги:
Отцы и Дети
Жанр книги:
Роман
Автор книги:
Тургенев
Издатель книги:
Русский вестник
Год издания книги:
1863
```

Рисунок 17 – Добавление книги в базу данных

При выборе второго пункта в дополнительном меню удаляется последний добавленный элемент.

При выборе третьего пункта в дополнительном меню пользователь вводит строку для поиска и программа выводит либо данные о книге, либо сообщение о неудаче (рисунок 18,19)

```
1. Добавить книгу
2. Удалить книгу
3. Поиск книги
4. Отобразить все книги
5. Сохранить базу данных в файл
6. Назад
>3
Введите строку для поиска:
Роман
Элемент был найден: Жизнь и смерть

Роман

Стефани Майер

-

2016

Для продолжения нажмите любую клавишу . . .
```

Рисунок 18 – удачный поиск

```
1. Добавить книгу
2. Удалить книгу
3. Поиск книги
4. Отобразить все книги
5. Сохранить базу данных в файл
6. Назад
>3
Введите строку для поиска:
Joe Barbagio
Элемент не был найден
Для продолжения нажмите любую клавишу . . .
```

Рисунок 19 – неудачный поиск

При выборе четвертого пункта в дополнительном меню программа выводит все данные о книгах (рисунок 20).

```
1. Добавить книгу
2. Удалить книгу
3. Поиск книги
4. Отобразить все книги
5. Сохранить базу данных в файл
6. Назад
>4
Жизнь и смерть
Роман
Стефани Майер
-
2016

Отцы и Дети
Роман
Тургенев
Русский вестник
1863

News 26.12.2017
-
Templar A.A
New York Times
2017

Для продолжения нажмите любую клавишу . . .
```

Рисунок 212 – Вывод данных о книгах

При выборе пятого пункта в дополнительном меню программа сохраняет базу данных в файл (рисунок 22).

```
1. Добавить книгу
2. Удалить книгу
3. Поиск книги
4. Отобразить все книги
5. Сохранить базу данных в файл
6. Назад
>5
Сохранение в файл произошло успешно
Для продолжения нажмите любую клавишу . . .
```

Рисунок 23 – Сохранение базы данных в файл

При выборе шестого пункта в дополнительном меню программа переходит к главному меню.

Заключение

При выполнении данной курсовой работы были получены навыки написания программ, использующих сложные структуры для реализации баз данных. Были освоены приемы создания пользовательского интерфейса в виде меню, изучены функции работы с консолью, способы обработки событий с клавиатуры. Были изучены функции для работы с файлами и получены базовые навыки программирования на языке си. Была освоена среда программирования Microsoft Visual Studio 2019. Также были получены навыки отладки программ с помощью различных инструментов среды Visual Studio.

Список использованных источников

1. Голицина О.Л, Максимов Н.В, Попов И.И Базы данных. - Москва: Форум, 2015. - 400 с.
2. Илюшечкин В.М Основы использования и проектирования баз данных. - Юрайт изд. - Москва: Учебник, 2015. - 214 с.
3. Основы программирования на языке СИ: учеб.пособие. - Тверь: Твер. гос. ун-т, 2017. - 159 с.
4. Миркин Б.Г Введение в анализ данных. Учебник и практикум. - Москва: Юрайт, 2015. - 176 с.
5. Свиридова М.Ю Система управления базами данных Access. - Москва: Академия, 2016. - 192 с.

Приложение А

Исходный код программы

Приложение А.1

Файл «Source.cpp»

```
#include <stdio.h>
#include <windows.h>
#include <locale.h>

#include "getVariant.h"
#include "printMainMenu.h"
#include "printMenu.h"

typedef struct list_
{
    int data;
    char name[100];
    char genre[100];
    char author[100];
    char izd[100];
    char year[100];
    struct list_* next;
} list;

int count=1;
int variant1, variant2;
char* fileName;
FILE* f;

list_* list_create(char* name1, char* genre1, char* author1, char* izd1,
char* year1)
{
    list* l;
    l = (list*)malloc(sizeof(list_));
    strcpy(l->name, name1);
    strcpy(l->genre, genre1);
    strcpy(l->author, author1);
    strcpy(l->izd, izd1);
    strcpy(l->year, year1);

    l->next = NULL;
    return l;
}

void list_print(list* head)
```

```

{
    list* curr = head;
    while (curr != NULL)
    {
        printf("%s%s%s%s\n", curr->name, curr->genre, curr->author,
curr->izd, curr->year);
        curr = curr->next;
    }
}

list *list_add_begin(list* head, char* name1, char* genre1, char* author1,
char* izd1, char* year1)
{
    list* append = list_create(name1, genre1, author1, izd1, year1);
    list* tmp = head;
    while (tmp->next != NULL)
    {
        tmp = tmp->next;
    }
    tmp->next = append;
    return head;
}

void list_delete(list * head)
{
    list* tmp = head;
    if (head == NULL)
    {
        printf("Невозможно удалить элемент - минимальный размер списка
- 1\n");
    }
    else
    {
        if (head->next == NULL)
        {
            free(head);
            head = NULL;
        }
        while (tmp->next->next != NULL)
        {
            tmp = tmp->next;
        }
        free(tmp->next);
        tmp->next = NULL;
    }
}

```

```

void list_search(list *head)
{
    list* curr = head;
    printf("Введите строку для поиска: \n");
    char string[100];
    fgets(string, sizeof(string), stdin);
    if (strcmp(string, curr->name) == 0 || strcmp(string, curr-
>genre) == 0 || strcmp(string, curr->author) == 0 || strcmp(string, curr-
>izd) == 0 || strcmp(string, curr->year) == 0)
    {
        printf("Элемент был найден: %s\n%s\n%s\n%s\n%s\n", curr-
>name, curr->genre, curr->author, curr->izd, curr->year);
    }
    else
    {
        printf("Элемент не был найден\n");
    }
}

```

```

void saveToFile(list *head)
{
    f = fopen(fileName, "w");
    if (!f)
    {
        printf("Файл не может быть открыт");
        exit(1);
    }

    list* curr = head;
    while (curr != NULL)
    {
        fprintf(f, "%s%s%s%s\n", curr->name, curr->genre, curr-
>author, curr->izd, curr->year);
        curr = curr->next;
    }

    printf("Сохранение в файл произошло успешно\n");
    fclose(f);
}

```

```

int main(int argc, char* argv[])
{
    setlocale(LC_ALL, "Rus");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    if (argc == 1)
    {
        printf("Ошибка - аргументы командной строки не заданы\n");
    }
}

```

```

        exit(1);
    }
    printf("База данных 'Список книг' разработана студентом Антоновым
Д.А.\n");
    Sleep(3000);
    fileName = argv[1];
    list* list_1 = NULL;
    char name2[100], genre2[100], author2[100], izd2[100], year2[100];
    int variant;
    do
    {
        printMainMenu();

        variant = get_variant(3);

        switch (variant)
        {
            case 1:
                printf("Введите первые элементы списка: \n");
                printf("Название книги: \n");
                fgets(name2, 100, stdin);
                printf("Жанр книги: \n");
                fgets(genre2, 100, stdin);
                printf("Автор книги: \n");
                fgets(author2, 100, stdin);
                printf("Издатель книги: \n");
                fgets(izd2, 100, stdin);
                printf("Год издания книги: \n");
                fgets(year2, 100, stdin);
                list_1 = list_create(name2, genre2, author2, izd2,
year2);

                do
                {
                    printMenu();
                    variant = get_variant(6);
                    switch (variant)
                    {
                        case 1:
                            printf("Название книги: \n");
                            fgets(name2, 100, stdin);
                            printf("Жанр книги: \n");
                            fgets(genre2, 100, stdin);
                            printf("Автор книги: \n");
                            fgets(author2, 100, stdin);
                            printf("Издатель книги: \n");
                            fgets(izd2, 100, stdin);
                            printf("Год издания книги: \n");
                            fgets(year2, 100, stdin);
                            list_1 = list_add_begin(list_1, name2, genre2,
author2, izd2, year2);

                            count++;
                            break;

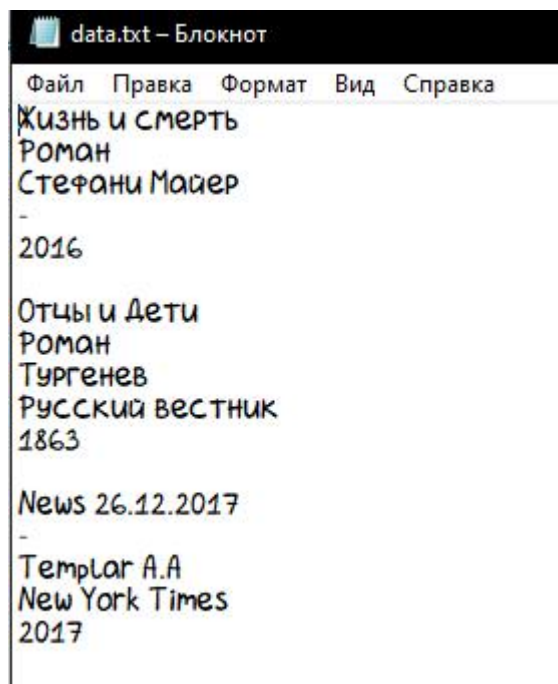
```

```

        case 2:
            list_delete(list_1);
            count--;
            break;
        case 3:
            list_search(list_1);
            break;
        case 4:
            list_print(list_1);
            break;
        case 5:
            saveToFile(list_1);
            break;
    }
    if (variant != 6)
    {
        system("pause");
    }
} while (variant != 6);
} while (variant != 2);
}

```

Приложение Б Содержимое файла



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

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту
по курсу «Программирование»
на тему «Разработка программы сложной структуры
методом нисходящего программирования»

29.11.22
Хер

Выполнил
студент группы 21ВО1

Антонов Д.А.

Принял

Митрохин М. А.

Пенза 2022

« » 2022 г.

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 202

Дата защиты проекта « 29 » октября 202

Руководитель Митрохин М.А.

Задание получил _____

« 1 » октября 202

Студент Антонов Даниил Алексеевич

Ант

Введени

1 Поста

1.1 Треб

2 Выбор

2.2 Опре

2.3 Опис

3 Отлад

4 Описан

4.1 Разра

4.2 Разра

5 Руково

Заключе

Список и

Приложе

Приложе