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

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе
по курсу «Программирование»
на тему «База данных»

Выполнил:

_____ Чабуев Р.П.

Принял:

_____ Патунин Д.В.
10.09.2022
удовл.

2022

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

Дата защиты проекта "10" 09. 2022

Руководитель Патунин Д.В.

Задание получил "25" марта 2022г.

Студент Чабуев Ростислав Павлович

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....9
4. Отладка и тестирование....12
5. Описание программы……12

5.1 Разработка приложения База_данных.exe....12

5.2 Разработка меню....14

5.3. Разработка функции на языке Ассемблер....14

6. Руководство пользователя....15

ЗАКЛЮЧЕНИЕ....19

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ....20

ПРИЛОЖЕНИЕ А ЛИСТИНГИ ПРОГРАММЫ....21

Приложение А.1 Файл nachalo.h....21

Приложение А.2 Файл nachalo.cpp……22

Приложение А.2 Файл kod.cpp....22.

ПРИЛОЖЕНИЕ В РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ, СНИМКИ ЭКРАНОВ, СОДЕРЖИМОЕ ФАЙЛОВ....33

Приложение В.1 Информационная заставка....33

Приложение В.2 Содержимое файла нен.txt....32

Введение

База данных - организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей.

База данных представляет собой упорядоченную информацию, объединенную в единое целое (например: список телефонных номеров, товаров, деталей, покупателей и т.д.).

Сам термин база данных появился в начале 1960-х годов, и был введён в употребление на симпозиумах, организованных компанией SDC в 1964 и 1965 годах, хотя понимался сначала в довольно узком смысле, в контексте систем искусственного интеллекта. В широкое употребление в современном понимании термин вошёл лишь в 1970-е годы.

С понятием базы данных тесно связано понятиесистемы управления базами данных(СУБД).СУБД– это комплекс программных средств, предназначенных для создания структуры новой базы данных, наполнения ее содержимым, редактирования содержимого и создания различных пользовательских представлений информации (отбор, сортировка, фильтрация и т.д.).

Создавая базу данных, пользователь стремится упорядочить информацию по каким-либо признакам и быстро извлекать выборку с производным сочетанием признаков. А сделать это можно, только если данные структурированы. Элементарная база данных представляется в виде двумерной таблицы.

В процессе выполнения курсовой работы была разработана программа на языке C++, позволяющая выполнять работу (поиск, сортировка, фильтрация, добавление и удаление строк и т.д.) с базой данных «Железнодорожные путешествия».

1. Постановка задачи

Необходимо разработать программу «База данных – Железнодорожные путешествия». Информация в базе данных структурирована по полям: ФИО, место отправления, место назначения, дата отъезда, длительность поездки. Для реализации данной программы необходимо определить структуру базы данных, размер, размер каждой записи базы данных.

Программа должна предоставить возможности: добавление новой записи, удаление записи; поиск, редактирование, сортировку и фильтрацию записей; сохранение базы данных в файл, загрузку базы данных из файла.

Для реализации 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. Это необходимо для четкого понимания функциональности программы.

Необходимо использовать сложные типы данных – массивы, структуры, файлы. Необходимо определиться с элементами управления. Необходимо изучить приемы и механизмы псевдографического интерфейса. Это необходимо для понимания, как и почему на консоли появляются псевдографические символы линии и т.д.

Устройство ввода – клавиатура. Необходимо научиться анализировать и различать события, возникающие от работы с клавиатурой. Необходимо однозначно идентифицировать и выполнять те или иные действия в зависимости от действий пользователя. Это необходимо для легкого использования программы.

Пользовательский интерфейс должен быть построен на основе меню. Необходимо изучить приемы программирования меню, выбрать подходящий вид меню. Это необходимо для создания интуитивно понятного интерфейса.

Многомодульность программы. Необходимо разделить программу на логические модули. Это повысит локализацию ошибок при отладке и тестировании программы.

Наличие заставки. Заставка-приветствие должна сообщать пользователю необходимую и достаточную информацию о программе.

2. Выбор решения

2.1 Определение необходимых модулей программы

Разработанная программа состоит из следующих модулей: «База_данных.exe» показана на рис.1.

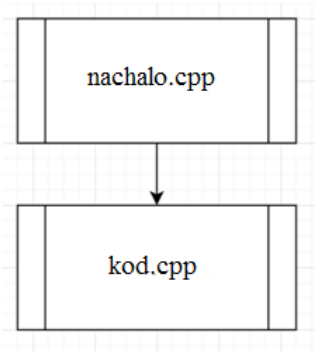


Рис. 1.

Файлы: «kod.cpp», «nachalo.h», «nachalo.cpp» - необходимы для включения функций обработки данных. Данные файлы содержат функции создания,удаления,чтения, поиска, редактирования, сортировки, фильтрации, вывода на экран базы данных и функцию сохранения результата обработки базы данных.

Приложение «База_данных.exe» представляет собой совокупность данных модулей.Приложение использует данные, содержащиеся в файле «heh». 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. 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, место отправления, место назначения, дата отъезда, длительность поездки.

2.2 Определение структуры базы данных

Для структурирования данных внутри файла была выбрана следующая структура данных:

```
typedef struct tagITEM
{
char strana[20];
char stolica[30];
char chisl[30];
char square[10];
  char dost[30];
  } ITEM;
```

Поля структуры: Страна, столица, численность страны, площадь, достопримечательность страны - имеют текстовый формат. Под Страну выделено 20 символов, под столицу 30, под площадь 10, а под длительность достопримечательность 30. Выбранная структура проста и соответствует всем требованиям программы.

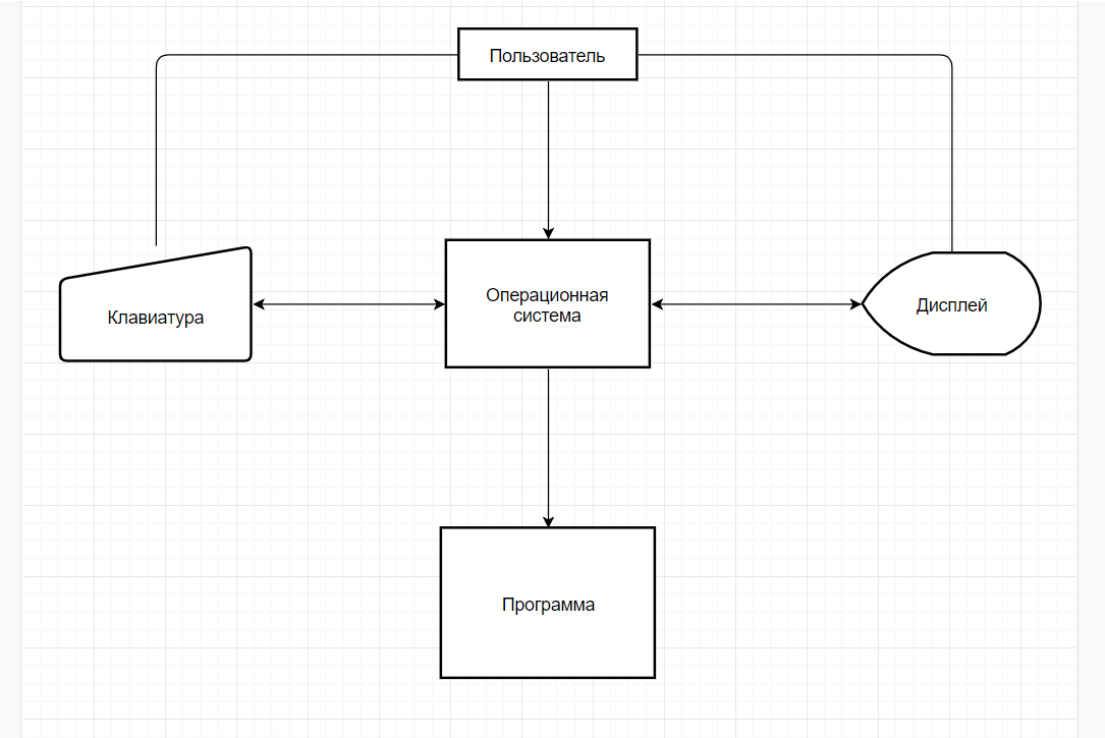


Рис.2

3. Описание разработки программы

Для написания данной программы был использован язык программирования Си.

В качестве среды программирования был выбран программный продукт VisualStudio 2010.

Интерфейс информационной системы представлен в виде меню, т.к. он является наиболее удобным. Исполнение навигации программы по пунктам меню будет осуществляться с помощью клавиатуры.

Чтобы пользователь мог пополнять базу данных будет предусмотрена функция «Добавить», а также функция «Удаление», дляудаление выбранной записи из базы данных.

Чтобы просмотреть полную информацию по какой-либо записи предусмотрена функция «Поиск».

Для завершения работы с программой и выхода в операционную систему будет предусмотрена функция «Выход из программы».

Разработка программы началась с реализации набора функций, необходимых для работы с файлом базы данных. Для чтения из файла входных данных используется функция Zag().Для сохранения файла были задействованы функция Soh(). Для поиска записей в содержимом по фамилии используется функция Poi().Для редактирования записей по полям используетсяфункция Red(). Для удаления записей по номеру записи используются функции Uda(). Для добавления записей по полям используется функция Dob().Для фильтрации по полям используются функция Fil(). Для сортировки по фамилии используется функция Upor(). Для вывода всей базы данных на экран используется функция Viv().

Основная программа выполнятся функцией main(). В функции main реализован цикл обработки событий. Также в функции предусмотрен вывод заставки и меню. Цикл обработки событий реагирует на события, вызванные клавиатурой, выполняет дешифрацию события и вызывает функции для обработки того или иного события. При активизации меню, управление переходит в функции реализации меню.

Был разработан алгоритм функции main. Блок схема функции представлена на рисунке. Функция отвечает за инициализацию консоли и за передачу управления в функции обработки меню. Цикл обработки событий главного окна обеспечивает активизацию меню и выход из него. После выхода из функции обработки меню функция обработки событий главного окна получает идентификатор выбранного пункта меню. В зависимости от идентификатора функция вызывает функцию обработки выбранного пункта меню.

Блок схема функции main() и функции обработки событий главного окна представлена на рисунке. Основной задачей данной функции является получения события от клавиатуры, если событие присутствует, то вызывается функция обработки

этого события.

На рисунке 3 представлена схема обработка событий.

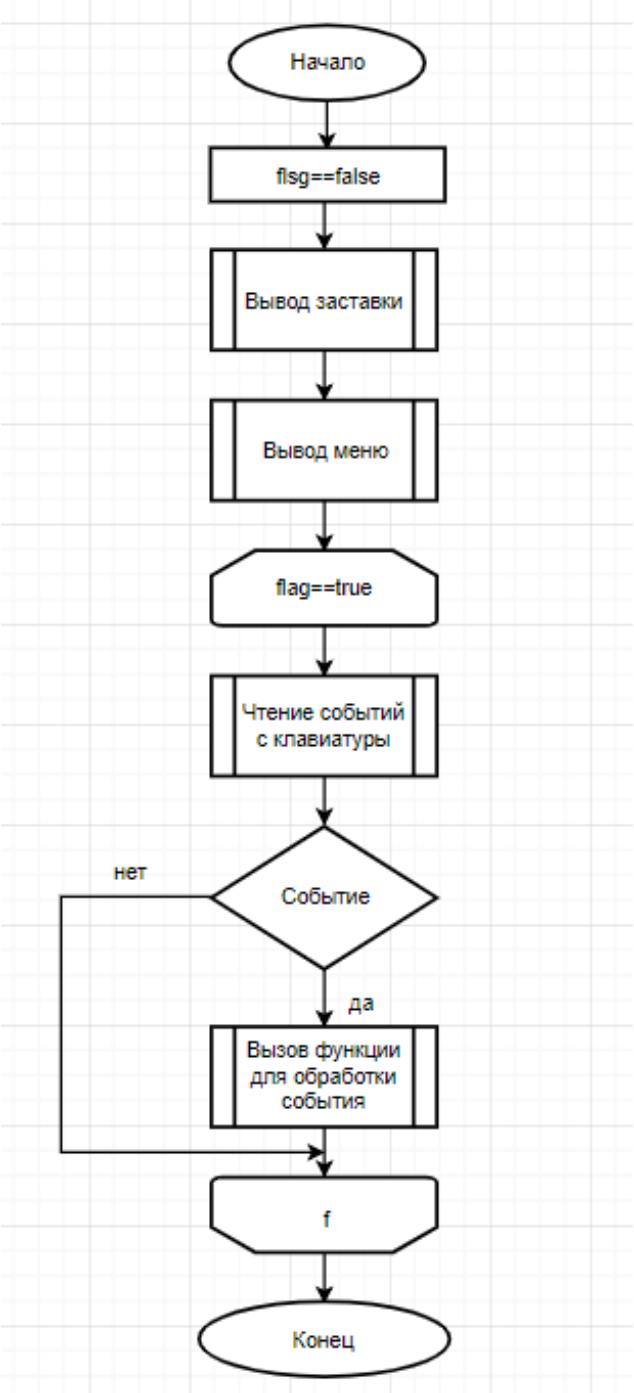


Рис.3

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

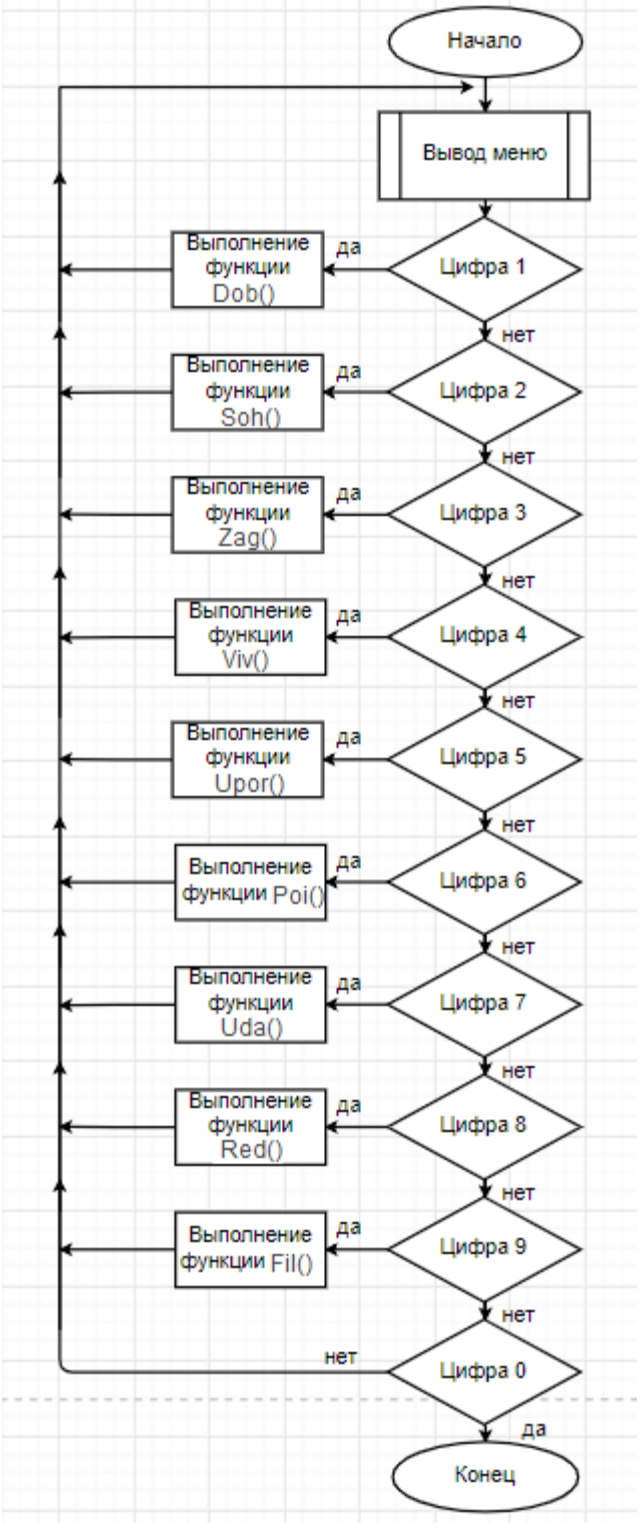


Рис. 4

Отладка и тестирование

В качестве среды разработки была выбрана программа VisualStudio2010. Программа предоставляет все средства необходимые при разработке и отладке разрабатываемых модулей и программ. Для отладки использовались такие инструменты как точка останова, выполнение кода по шагам, анализ содержимого локальных и глобальных переменных, анализ содержимого памяти.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы. В ходе тестирования было выявлено и исправлено множество проблем, связанных с перерисовкой содержимого таблицы, с перерисовкой меню и многое другое. Также были добавлены проверки, связанные с недопустимостью обработки некоторых пунктов меню при определенных условиях.

4. Описание программы

5.1 Разработка приложения База_данных.exe

Приложение «База_данных.exe» является основным модулем программы. При запуске программы происходит вывод заставки-приветствия, после чего программа готова к выполнению своих функций. После заставки происходит активация меню, затем, выбрав соответствующую опцию, программа переходит к выполнению соответствующих функций. Выход из меню осуществляется после выбора одного из пунктов меню. Описание всех состояний выполнено в таблице 1.

Таблица 1 – описание состояний главного меню

Клавиша, вызывающая	Опция	Действия программы
1	Загрузка из файла	Производится загрузка базы данных из файла
2	Сохранение в файл	Производится сохранение базы данных в файл
3	Вывод на экран	Производится вывод базы данных на экран
4	Редактирование записей	Производится редактирование записей
5	Добавление записи в базу	Добавление записи
6	Удаление записей	Производится удаление
7	Сортировка записей	Производится сортировка
8	Фильтрация записей	Производится фильтрация записей
9	Поиск записей	Производится поиск записи
0	Выход	Выход из программы

На рисунке 5 изображена иерархическая структура программы.

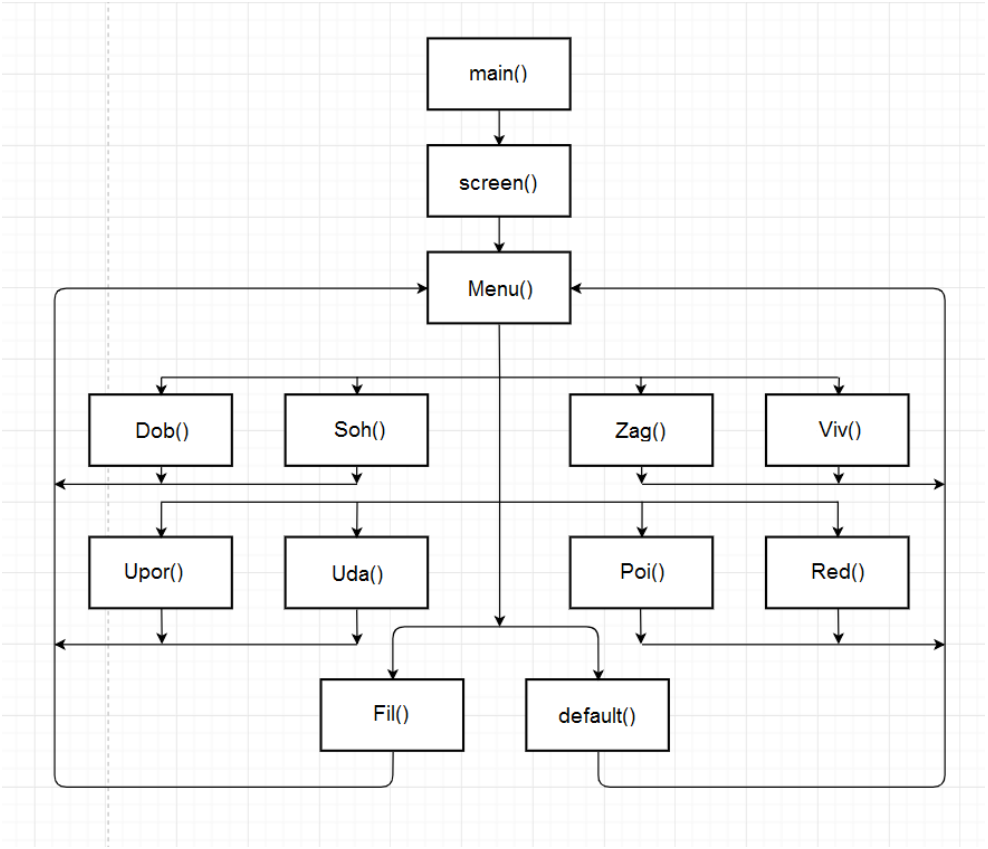


Рис. 5

5.2 Разработка меню

Меню реализовано в файле «kod.cpp» и состоит из следующих пунктов:

Создание новой базы данных, Сохранение базы данных в файл, Загрузка базы данных из файла, Вывод базы данных на экран, Редактирование базы, Удаление записей, Сортировка записей, Поиск записей, Фильтрация записей, Выход.

Меню реагирует на нажатие клавиш: цифр от 0 до 9. При этом на выбранном пункте меню управление передается на соответствующую функцию.

5.3. Разработка функции на языке Ассемблер

Как известно, функции, реализованные на языке ассемблера, работают быстрее и занимают меньше места. В качестве разрабатываемой функции была выбрана функция инкрементирования, а также запуск меню программы.

Описание функции выполнено на языке С, а тело функции представляет собой ассемблерную вставку, компилятор, входящий в состав среды разработки VS2010-2019, позволяет это делать. Для этого перед ассемблерной вставкой необходимо написать ключевое слово “_asm ”, а сам код поместить в фигурные скобки.

Для реализации функций вывода были использованы следующие ассемблерные инструкции:

— инструкция push выполняет «вталкивание» используемых регистров в стек;

— инструкция mov выполняет копирование содержимого источника в приемник.

— инструкция add суммирование содержимого источника и приемника

— инструкция pop выполняет «выталкивание» используемых регистров из стека.

Код функции представлен в приложении.

5. Руководство пользователя

Программа «База_данных.exe» предназначена для хранения информации о странах мира. Программа имеет интуитивно понятный интерфейс и поддерживает такие операции как добавление записи, удаление, редактирование, сортировку, фильтрацию, поиск по записям, загрузку базы данных из файла и сохранение в файл. Показано на рисунке 6.

Рис. 6

Для загрузки базы данных из файла необходимо выбрать пункт меню «загрузить». При успешном выполнении операции будет выведено соответствующее сообщение.

Показано на рисунке 7.

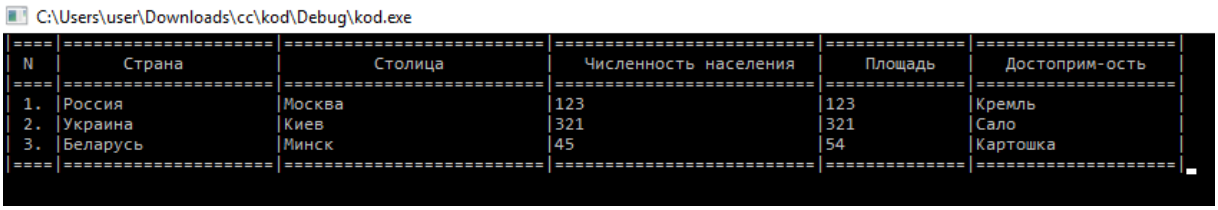


Рис. 7

Для вывода всей базы данных на экран необходимо выбрать пункт меню «вывести».

Для добавления записи в базу данных необходимо выбрать пункт «добавить». После нужно ввести необходимые данные, после ввода каждой строки нажимать клавишу Enter. Запись будет добавлена в базу данных, и на экран будет выведено меню.

Показано на рисунке 8.

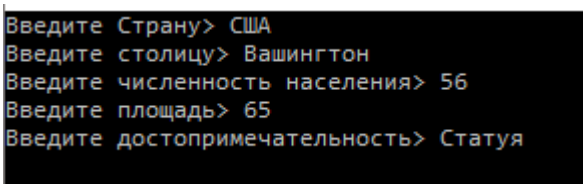


Рис. 8

Для удаления записи из базы данных следует выбрать пункт «удаление».

После ввода номера записи она будет удалена из базы данных, и будет выведено сообщение «Удалено». Показано на рисунке 9.

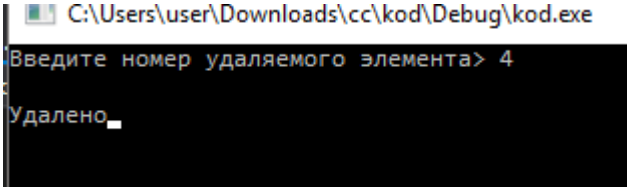


Рис. 9

Для редактирования записи необходимо выбрать пункт «редактирование». Для поиска записи, которую необходимо отредактировать нужно ввести фамилию. Затем необходимо ввести новые данные. Показано на рисунке 10.

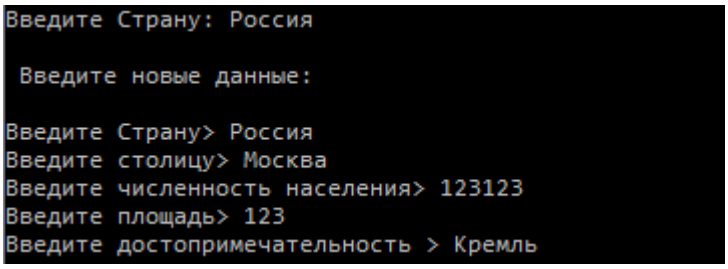


Рис. 10

Для поиска нужной записи базы данных нужно выбрать пункт «поиск». Показано на рисунке 11.

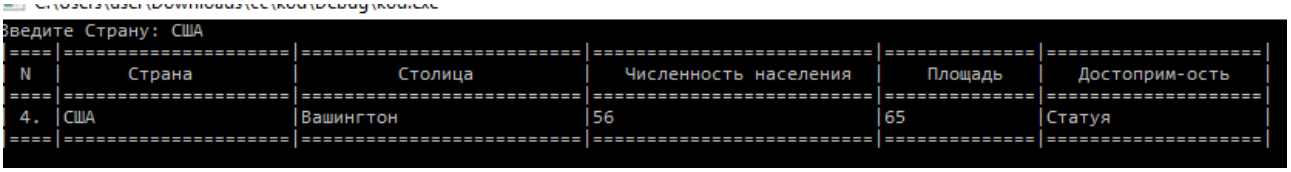


Рис. 11

Для сортировки нужно выбрать пункт «Сортирока». После чего записи в базе данных будут отсортированы по фамилии. Показано на рисунке 12.

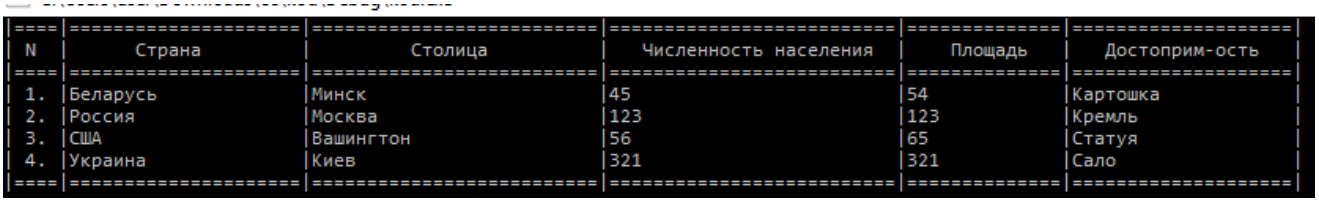


Рис.12

Для фильтрации записей необходимо выбрать пункт «фильтрация». На экране появится меню фильтрации. Показано на рисунке 13.

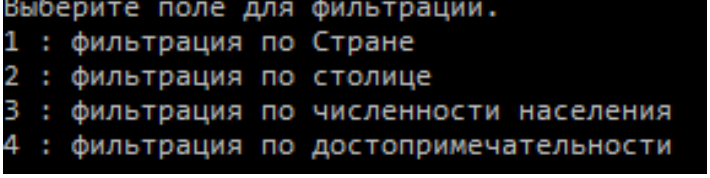


Рис. 13

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 С и Ассемблер. Изучены основные возможности среды программирования VisualStudio 2010 - 2019. Получены навыки отладки и тестирования программ.

В рамках выполнения курсовой работы была написана программа, необходимая для ведения базы данных «Страны мира». Программа предоставляет небольшой, но достаточный для использования список возможностей. Не требует большого количества системных ресурсов и обеспечивает быструю обработку базы данных.

Список используемых источников

1. Герберт Шилдт. Полный справочник по С.
2. Н. Культин. Основы программирования в MicrosoftVisual C++ 2010, 2010г
3. Харви и Пол Дейтел. Как программировать на С.
4. Александр Чиртик. Программирование на С++. Трюки и эффекты, 2010г.
5. http://cyberforum.ru/
6. https://habr.com/ru

Приложение А

Листинги программы

Приложение А.1
Файл nachalo.h

```
#pragma once
void screen( void );
void Menu( void );
```

Приложение А.2
Файл nachalo.cpp

```
#include<stdio.h>
#include <windows.h>
#include "StdAfx.h"

char vip[] = "0 : Выход из программы\n";
char zvbd[] = "1 : Загрузить в БД\n";
char sbd[] = "2 : Сохранить БД\n";
char vbdne[] = "3 : Вывести БД на экран\n";
char rbd[] = "4 : Редактировать БД\n";
char dvbd[] = "5 : Добавить в БД\n";
char uizbd[] = "6 : Удалить из БД\n";
char sort[] = "7 : Сортировка\n";
char filtr[] = "8 : Фильтровка\n";
char search[] = "9 : Поиск в БД\n";
char str[] = ">";
void screen(void) {

printf("_____\n");
	printf("|
|\n");
	printf("|
|\n");
	printf("|                                Курсовая работа
|\n");
	printf("|                                      по
|\n");
	printf("|                                программированию
|\n");
	printf("|
_____|\n");
	printf("|                        На тему: База данных. Столицы мира
|\n");
	printf("|
_____|\n\n\n\n");
	printf("Выполнил студент группы 21ВВ3:\n");
	printf("Чабуев Р.П.\n");
	Sleep(3000);
	system("cls");
}

void Menu(void) {
	_asm
	{
		lea eax, vip
		push eax
		call printf
		pop eax

		lea eax, zvbd
		push eax
		call printf
		pop eax


		lea eax, sbd
		push eax
		call printf
		pop eax
```

```
lea eax, vbdne
push eax
call printf
pop eax

lea eax, rbd
push eax
call printf
pop eax

lea eax, dvbd
push eax
call printf
pop eax

lea eax, uizbd
push eax
call printf
pop eax

lea eax, sort
push eax
call printf
pop eax

lea eax, filtr
push eax
call printf
pop eax

lea eax, search
push eax
call printf
pop eax

lea eax, str
push eax
call printf
pop eax
```

Приложение А.2
Файл kod.cpp

```
#include "stdafx.h"

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<string.h>

#include<tchar.h>

#include<windows.h>

#include"nachalo.cpp"

#include"nachalo.h"

#include<locale.h>

typedef struct tagITEM

{

char strana[20];

char stolica[30];

char chisl[30];

char square[10];

char dost[30];

} ITEM;

ITEM Men[20];

int Number;
```

```
FILE *F;

struct tagITEM *p;

void Zag( void )

{

int i=0;

char strana[50], stolica[50],chisl[50], square[50], dost[50];

if ((F = fopen("database.txt", "r")) == NULL)

{

fprintf(stderr, "Невозможно открыть для чтения файл 'database.txt'\n");

return;

  }

else{

while (!feof(F)){

 if ((p = (struct tagITEM*)(malloc(sizeof(struct tagITEM)))) == NULL)

             {

                   printf("Не удалось загрузитть данные\n");

                   break;

    }

 else{

fscanf(F,       "\"%[^\"]\"\"%[^\"]\"\"%[^\"]\"\"%[^\"]\"\"%[^\"]\"\n",       &(strana),
&(stolica), &(chisl), &(square), &(dost));

                   if (strcmp(strana, "") != 0)

                   {
```

```
                        strcpy(Men[Number].strana, strana);
                        strcpy(Men[Number].stolica, stolica);
                        strcpy(Men[Number].chisl, chisl);
                        strcpy(Men[Number].square, square);
                        strcpy(Men[Number].dost, dost);
                        i++;
                        Number++;
                    }
   }
   }
if (i != 0){
        printf("Успешно");
        fclose(F);
   }
}
}
void Soh( void )
{
int i;
if ((F = fopen("database.txt", "w")) == 0)
printf("Невозможно открыть/создать для записи файл 'database.txt'\n");
else{
for (i = 0; i < Number; i ++){
```

```
fprintf(F,     "\"%s\"\"%s\"\"%s\"\"%s\"\"%s\"\n",     Men[i].strana,     Men[i].stolica,
Men[i].chisl, Men[i].square, Men[i].dost);

  }

printf("Успешно");

fclose(F);

  }

}

void Viv( void )

{

int i;

printf("|====|====================|=========================|
=========================|=============|====================|\n"

    "| N  |      Страна         |         Столица          |   Численность населения  |
Площадь   |   Достоприм-ость   |\n"

"|====|====================|=========================|=========================|
=============|====================|\n");

for (i = 0; i < Number; i ++)

printf("|%2i.  |%-20s  |%-25s  |%-25s  |%-13s  |%-20s|\n",  i  +  1,  Men[i].strana,
Men[i].stolica, Men[i].chisl, Men[i].square, Men[i].dost);

printf("|====|====================|=========================|
=========================|=============|====================|");

}

void Red( void ){

char f[20];

 int n=0;

 printf("Введите Страну: ");
```

```
 scanf("%s", &f);

 for(int i=0; i<Number; i++){

       if(strcmp(f, Men[i].strana)==0){

printf("\n Введите новые данные:\n\n");

printf("Введите Страну> ");

scanf("%s", Men[i].strana);

printf("Введите столицу> ");

scanf("%s", Men[i].stolica);

printf("Введите численность населения> ");

scanf("%s", &Men[i].chisl);

printf("Введите площадь> ");

scanf("%s", &Men[i].square);

printf("Введите достопримечательность > ");

scanf("%s", &Men[i].dost);

             n++;

 printf("|====|====================|==========================|
==========================|=============|====================|\n"

                   "|  N   |        Страна            |            Столица             |
Численность населения  |    Площадь   |   Достоприм-ость   |\n"

                   "|====|====================|==========================|
==========================|=============|====================|\n");

printf("|%2i.  |%-20s  |%-25s  |%-25s  |%-13s  |%-20s|\n",  i  +  1,  Men[i].strana,
Men[i].stolica, Men[i].chisl, Men[i].square, Men[i].dost);

                        printf("|====|====================|==========================|
==========================|=============|====================|");

             break;
```

```
       }
 }
 if(n==0)
       printf("\nДанного страны в базе нет");
}
void Dob( void )
{
if (Number == 19)
{
fprintf(stderr, "Невозможно добавить больше\n");
return;
  }
printf("Введите Страну> ");
scanf("%s", Men[Number].strana);
printf("Введите столицу> ");
scanf("%s", Men[Number].stolica);
printf("Введите численность населения> ");
scanf("%s", &Men[Number].chisl);
printf("Введите площадь> ");
scanf("%s", &Men[Number].square);
printf("Введите достопримечательность> ");
scanf("%s", &Men[Number].dost);
  Number ++;
```

```
}

void Uda( void )

{

int i;

printf("Введите номер удаляемого элемента> ");

scanf("%i", &i);

if (i < 1 || i > Number)

{

fprintf(stderr, "Элемент с номером %i не существует\n", i);

return;

  }

for (i --; i < Number - 1; i ++)

    Men[i] = Men[i + 1];

  Number --;

printf("\nУдалено");

}

void Upor( void )

{

int i, j;

  ITEM Temp;

for (j = Number - 1; j > 0; j --)

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

if (strcmp(Men[i].strana, Men[i + 1].strana) > 0)
```

```
      {

        Temp = Men[i];

        Men[i] = Men[i + 1];

Men[i + 1] = Temp;

      }

printf("|====|====================|=========================|
=========================|=============|====================|\n"

 "| N  |      Страна          |          Столица          |   Численность населения  |
Площадь   |   Достоприм-ость   |\n"

 "|====|====================|=========================|=========================|
=============|====================|\n");

for (i = 0; i < Number; i ++)

printf("|%2i.  |%-20s  |%-25s  |%-25s  |%-13s  |%-20s|\n",  i  +  1,  Men[i].strana,
Men[i].stolica, Men[i].chisl, Men[i].square, Men[i].dost);

                          printf("|====|====================|=========================|
=========================|=============|====================|");

}

void Fil( void ){

 char f[20];

 int n=0;

 int g;

 printf("Выберите поле для фильтрации.\n");

 printf("1 : фильтрация по Стране \n");

 printf("2 : фильтрация по столице \n");

 printf("3 : фильтрация по численности населения \n");

 printf("4 : фильтрация по достопримечательности \n");
```

```
 scanf("%d", &g);

 printf("\n");

 switch (g)

 {

 case 1:

 printf("Введите Страну: ");

 scanf("%s", &f);

 for(int i=0; i<Number; i++){

       if(strcmp(f, Men[i].strana)==0){

             printf("|====|====================|=========================|
=========================|=============|====================|\n"

             "| N  |      Страна          |          Столица           |   Численность
населения  |    Площадь   |   Достоприм-ость   |\n"

 "|====|====================|=========================|=========================|
=============|====================|\n");

printf("|%2i.  |%-20s  |%-25s  |%-25s  |%-13s  |%-20s|\n",  i  +  1,  Men[i].strana,
Men[i].stolica, Men[i].chisl, Men[i].square, Men[i].dost);

printf("|====|====================|=========================|
=========================|=============|====================|");

             n++;

       }

 }

 if(n==0)

             printf("\nСтраны с данным именем в базе нет");

 break;

 case 2:
```

```
      printf("Введите столицу: ");

      scanf("%s", &f);

      for(int i=0; i<Number; i++){

            if(strcmp(f, Men[i].stolica)==0){

      printf("|====|====================|=========================|
    =========================|=============|====================|\n"

                        "|  N   |        Страна            |            Столица             |
    Численность населения  |    Площадь   |   Достоприм-ость   |\n"

      "|====|====================|=========================|=========================|
=============|====================|\n");

    printf("|%2i.  |%-20s  |%-25s  |%-25s  |%-13s  |%-20s|\n",  i  +  1,  Men[i].strana,
    Men[i].stolica, Men[i].chisl, Men[i].square, Men[i].dost);

    printf("|====|====================|=========================|
    =========================|=============|====================|");

                  n++;

            }

      }

      if(n==0)

                  printf("\nСтраны с такой столицей в базе нет");

      break;

      case 3:

      printf("Введите достопримечательность: ");

      scanf("%s", &f);

      for(int i=0; i<Number; i++){

            if(strcmp(f, Men[i].chisl)==0){
```

```
 printf("|====|====================|=========================|
=========================|=============|====================|\n"

                    "|  N   |        Страна            |           Столица            |
Численность населения  |    Площадь    |   Достоприм-ость   |\n"

 "|====|====================|=========================|=========================|
=============|====================|\n");

printf("|%2i.  |%-20s  |%-25s  |%-25s  |%-13s  |%-20s|\n",  i  +  1,  Men[i].strana,
Men[i].stolica, Men[i].chisl, Men[i].square, Men[i].dost);

printf("|====|====================|=========================|
=========================|=============|====================|");

            n++;

      }

 }

 if(n==0)

            printf("\nДанной строки в базе нет");

 break;

 case 4:

 printf("Введите достопримечательность: ");

 scanf("%s", &f);

 for(int i=0; i<Number; i++){

      if(strcmp(f, Men[i].dost)==0){

 printf("|====|====================|=========================|
=========================|=============|====================|\n"

                    "|  N   |        Страна            |           Столица            |
Численность населения  |    Площадь    |   Достоприм-ость   |\n"

                    "|====|====================|=========================|
=========================|=============|====================|\n");
```

```
printf("|%2i.  |%-20s  |%-25s  |%-25s  |%-13s  |%-20s|\n",  i  +  1,  Men[i].strana,
Men[i].stolica, Men[i].chisl, Men[i].square, Men[i].dost)

                        printf("|====|====================|=========================|
=========================|=============|====================|");

            n++;

      }

 }

 if(n==0)

            printf("\nДанной строки в базе нет");

 break;

 default:

      printf("\nВведено неверное число.");

      break;

 }

}

void Poi( void ){

 char f[20];

 int n=0;

 printf("Введите Страну: ");

 scanf("%s", &f);

 for(int i=0; i<Number; i++){

      if(strcmp(f, Men[i].strana)==0){

 printf("|====|====================|=========================|
=========================|=============|====================|\n"
```

```
       "| N  |      Страна          |         Столица          |   Численность населения
|    Площадь   |   Достоприм-ость   |\n"        "|====|======================|
=========================|=========================|=============|
====================|\n");

printf("|%2i.  |%-20s  |%-25s  |%-25s  |%-13s  |%-20s|\n",  i  +  1,  Men[i].strana,
Men[i].stolica, Men[i].chisl, Men[i].square, Men[i].dost);

printf("|====|======================|=========================|
=========================|=============|====================|");

             _asm{

                   push eax

                   mov eax,n

                   add eax,1

                   mov n,eax

                   pop eax

             };

       }

 }

 if(n==0)

             printf("Данной страны в базе нет");

}




void main(void)

{

 system("chcp 1251");

int c;
```

```
screen();
  Number = 0;
do{
 Menu();
 fflush(stdin);
   c=getch();
 printf("\n");
switch (c)
    {
case'0':
exit(0);
       break;
case'1':
 system("cls");
 Zag();
 _getch();
 system("cls");
 break;

case'2':
system("cls");
 Soh();
 _getch();
```

```
 system("cls");
 break;
case'3':
 system("cls");
Viv();
_getch();
 system("cls");
 break;
case'4':
 system("cls");
Red();
   _getch();
 system("cls");
break;
case'5':
 system("cls");
Dob();
   _getch();
 system("cls");
break;
case'6':
 system("cls");
Uda();
```

```
   _getch();
 system("cls");
break;
case'7':
 system("cls");
Upor();
   _getch();
 system("cls");
break;
case'8':
 system("cls");
Fil();
   _getch();
 system("cls");
break;
case'9':
 system("cls");
Poi();
   _getch();
 system("cls");
break;
 default:
       {
```

```
        system("cls");

        printf("\n Введен недопустимый символ\n");

        _getch();

        system("cls");

        break;

        }

        }

 } while (c != 0);

}
```

Приложение В

Результаты работы программы, снимки экрана, содержимое файлов.

Приложение В.1 Информационная заставка, рисунок 14.

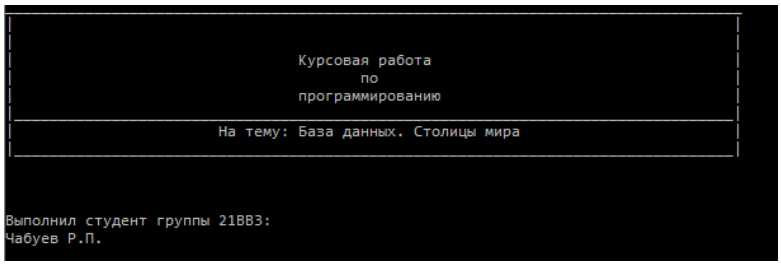


Рис. 14

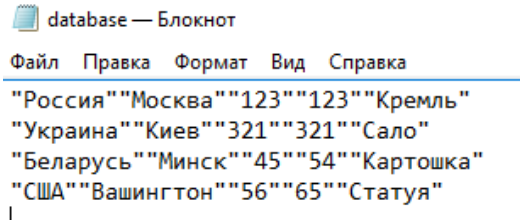
Приложение В.2 Содержимое файла data.txt на рисунке 15.

N	Страна	Столица	Численность населения	Площадь	Достоприм-ость
1.	Россия	Москва	123	123	Кремль
2.	Украина	Киев	321	321	Сало
3.	Беларусь	Минск	45	54	Картошка
4.	США	Вашингтон	56	65	Статуя

Рис. 15

"Россия""Москва""123""123""Кремль"

"Украина""Киев""321""321""Сало"

"Беларусь""Минск""45""54""Картошка"

"США""Вашингтон""56""65""Статуя"