ГУАП

КАФЕДРА №34

КУРСОВАЯ РАБОТА  
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | К. А. Жиданов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ |
| БАЗА ДАННЫХ «КАЛЕНДАРЬ» |
| по дисциплине: ПРОГРАММИРОВАНИЕ. МЕТОДЫ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛИ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТЫ  ГР. № | 3643 |  |  |  | А.А. Прохорова А.Д. Юницкая |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2018

**Содержание**

ВВЕДЕНИЕ…………………………………………………………………..……3

1. Техническое задание……………………………………….…………..….3
2. Структура данных………………………………………………………….………….8
3. Схема алгоритма……………………………………………………….…8
4. Текст программы………………………………………………………….17
5. Инструкция по применению……………………….…………………….24
6. Спецификация программы……………………………………………....26

ЗАКЛЮЧЕНИЕ……………………………………………………………..….27

СПИСОК ЛИТЕРАТУРЫ……………………………………………………….27

ВВЕДЕНИЕ

При современном темпе жизни люди начали забывать, что и когда у них запланировано. На прилавках магазина в различных формах продается огромное количество планеров. Но что делать человеку, который не бегает между дел, а работает за компьютером? В таком случае электронная база «Календарь» может помочь.

Разработанный программный продукт предназначен для вывода содержимого электронной базы «Календарь» по месяцам, вывода по фамилиям и категориям.

Системные требования:

* Процессор с тактовой частотой 1,6 ГГц или большей;
* 1 ГБ ОЗУ;
* 2 Мб доступного пространства на жестком диске;
* ОС Windows;
* Стандартный видеоадаптер.

1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

**1. Введение**

* 1. **Наименование программы**

База данных «Календарь».

**Основания для разработки**

В качестве языка используется C++, а в качестве среды разработки - используется Code Blocks. Именно в этой программе пишется код и проверяется на наличие ошибок и исправность работы.

* 1. **Назначение и область применения**

Программа предназначена для вывода содержимого электронной базы «Календарь» по месяцам, вывода по фамилиям и категориям.

**2. Требования к программе**

**2.1. Требования к источникам информации**

Источниками информации для Календаря является вводимая информация пользователем согласно инструкции по применению (см. в курсовой работе).

**2.2. Требования к предоставлению информации**

«Текстовой выход» не требует красивого дизайна, нужна только информация, пример на Рис.1:

2.2.1. Команды построчно, с индивидуальным номером и белым цветом.

2.2.2. Каждый блок через пустую строку.

2.2.3. Красным цветом внесенная информация в календарь.

|  |
| --- |
| C:\... |
| Выберите тип даты (праздник/событие): Celebration  Celebration Nastya Birthday 23 September 1997  Выберете:  1 – добавить запись  2 – удалить запись  3 – показать записи по типу даты (праздник/событие)  4 – показать записи по фамилиям  5 – показать записи по месяцу  0 – выход  > |

Рис.1. Пример вывода информации.

**2.3. Требования к функциональным характеристикам**

Программа «Календарь» должна обеспечивать возможность выполнения перечисленных ниже свойств:

2.3.1. Пополнение базы (дни рождения, праздники, события).

2.3.2. Редактирование базы.

2.3.3. Удаление записей.

2.3.4. Вывод календаря на текущий месяц с указанием праздников, дней рождения и событий.

2.3.5. Вывод дат либо по категориям, либо по фамилиям.

**2.4. Требования к надежности**

**2.4.1. Требования к обеспечению надежного функционирования программы**

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

1. использованием лицензионного программного обеспечения;
2. регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов;
3. организацией бесперебойного питания технических средств.

**3. Условия эксплуатации**

**3.1. Климатические условия эксплуатации**

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

**3.2. Требования к квалификации и численности персонала**

Программа предназначена помогать людям, записывая нужные мероприятия: дополнительный персонал не требуется. Квалификация пользователя – уровень пользователь.

**3.3. Требования к исходным кодам и языкам программирования**

Для написания программы был выбран язык программирования С++.

**3.3.1 Требования к защите информации и программ**

Требования к защите информации и программ не предъявляются.

**4. Требования к программной документации**

**4.1. Предварительный состав программной документации**

Состав программной документации должен включать в себя:

1. техническое задание;

2. текст курсовой работы, поясняющий и раскрывающий суть программы.

**5. Стадии и этапы разработки**

**5.1. Стадии разработки**

Разработка должна быть проведена в три стадии:   
1. разработка технического задания;   
2. рабочее проектирование;   
3. внедрение.

**4.2. Этапы разработки**

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.   
На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. разработка программы;
2. разработка программной документации;
3. испытания программы.

На стадии внедрения должен быть выполнен этап разработки, подготовка и передача программы.

**4.3. Содержание работ по этапам**

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:   
1. постановка задачи;   
2. определение и уточнение требований к техническим средствам;   
3. определение требований к программе;  
4. определение стадий, этапов и сроков разработки программы и документации на неё;   
5. согласование и утверждение технического задания.   
 На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.   
 На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.   
 На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:   
1. разработка, согласование и утверждение и методики испытаний;   
2. проведение приемо-сдаточных испытаний;   
3. корректировка программы и программной документации по результатам испытаний.   
 На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика.

1. СТРУКТУРА ДАННЫХ

Структура базы данных заполняется из текстового файла «kalendar.txt», содержащего данные о датах: Тип даты (праздник/событие), фамилия, название самого праздника или события, число, месяц, год.

char date\_type[50]. Символьный массив для хранения типа даты (праздника/события). Максимальное число символов 49.

char familia[50]. Символьный фамилий. Максимальное число символов 49.

char name[50]. Символьный массив для хранения самих названий праздников или событий.

int data. Переменная целого типа, хранящая число.

char month[50]. Символьный массив, хранящий месяц.

int year. Переменная целого типа, хранящая год.

При переполнении массивов программа будет работать некорректно.

1. СХЕМА АЛГОРИТМА

3.1. Словесное описание.

1. Начало.
2. При запуске программы происходит заполнение структуры данных из файла.
3. После выполнения предыдущего пункта на экране будет предложено выполнить одно из шести возможных действий. После ввода соответствующей цифры будет выполнено описанное действие.
4. При вводе цифры 1 (добавить запись) запустится функция «func\_dobvleniya\_dat», которая после выполнения указаний, отображённых на экране, внесёт данные в базу. После этого произойдёт перезаполнение структуры.
5. При вводе цифры 2 (удалить запись) будет предложено ввести само название праздника или события и фамилию, после чего файл перезапишется без данных об указанном пользователем празднике/событии.
6. При вводе цифры 3 (показать записи по типу даты (праздник/событие)) на экране появится запрос с требованием указать тип даты (праздник/событие). После ввода программа выполнит поиск дат по запрошенному типу даты(празднику/событию) и выведет результат на экран.
7. При вводе цифры 4 (показать записи по фамилии) на экране появится запрос с требованием указать фамилию. После ввода программа выполнит поиск дат по запрошенной фамилии и выведет результат на экран.
8. При вводе цифры 5 (показать записи по месяцу) на экране появится запрос с требованием указать месяц. После ввода программа выполнит поиск дат по запрошенному месяцу и выведет результат на экран.
9. После выполнения любого с 1 по 5 действие будет предложено повторное выполнение любого из 6 действий. В случае ввода нуля будет осуществлён выход из программы.
10. Конец.

3.2. Схемы алгоритма.

3.2.1. Схема алгоритма функции «compereString».

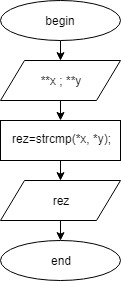


Рис. 1 - Схема алгоритма функции «compereString».

3.2.2. Схема алгоритма функции «func\_dobvleniya\_dat».

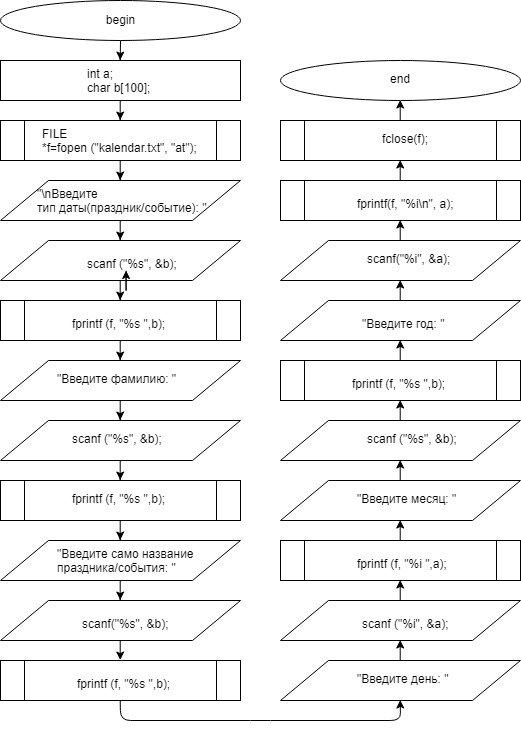


Рис. 2. Схема алгоритма функции «func\_dobvleniya\_dat».

3.2.3 Схема алгоритма участка основной программы «case 1».

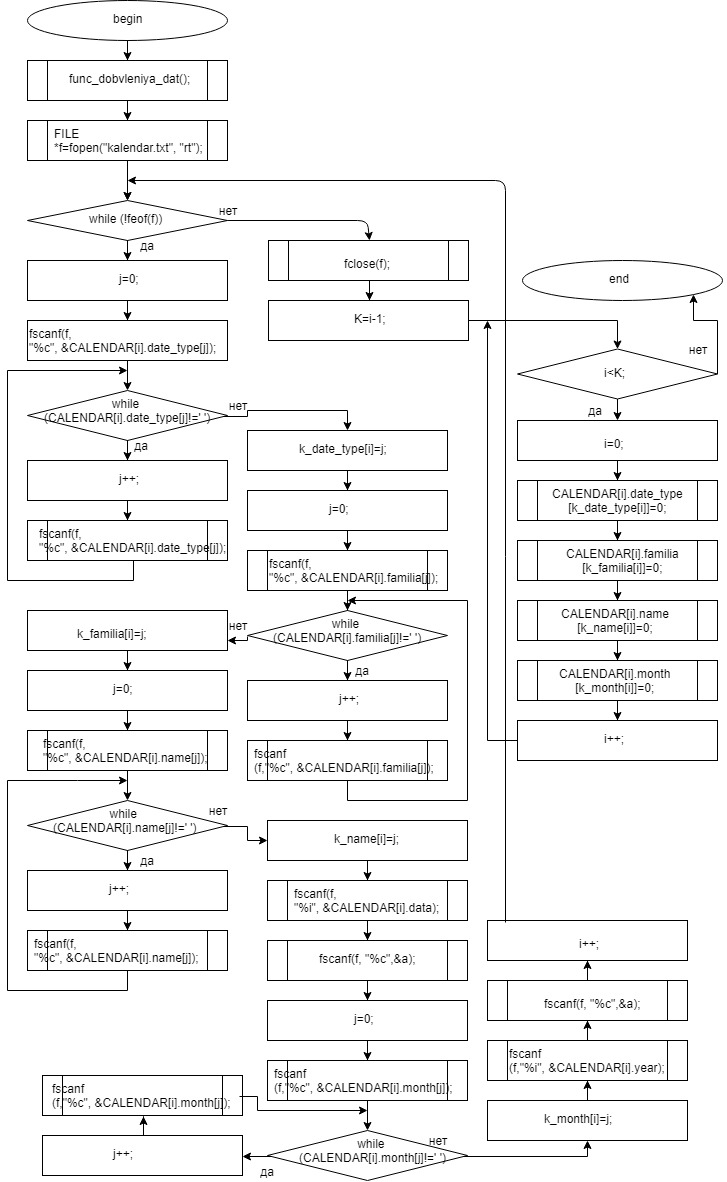


Рис.3. Схема алгоритма участка основной программы «case 1».

3.2.4. Схема алгоритма участка основной программы «case 2».

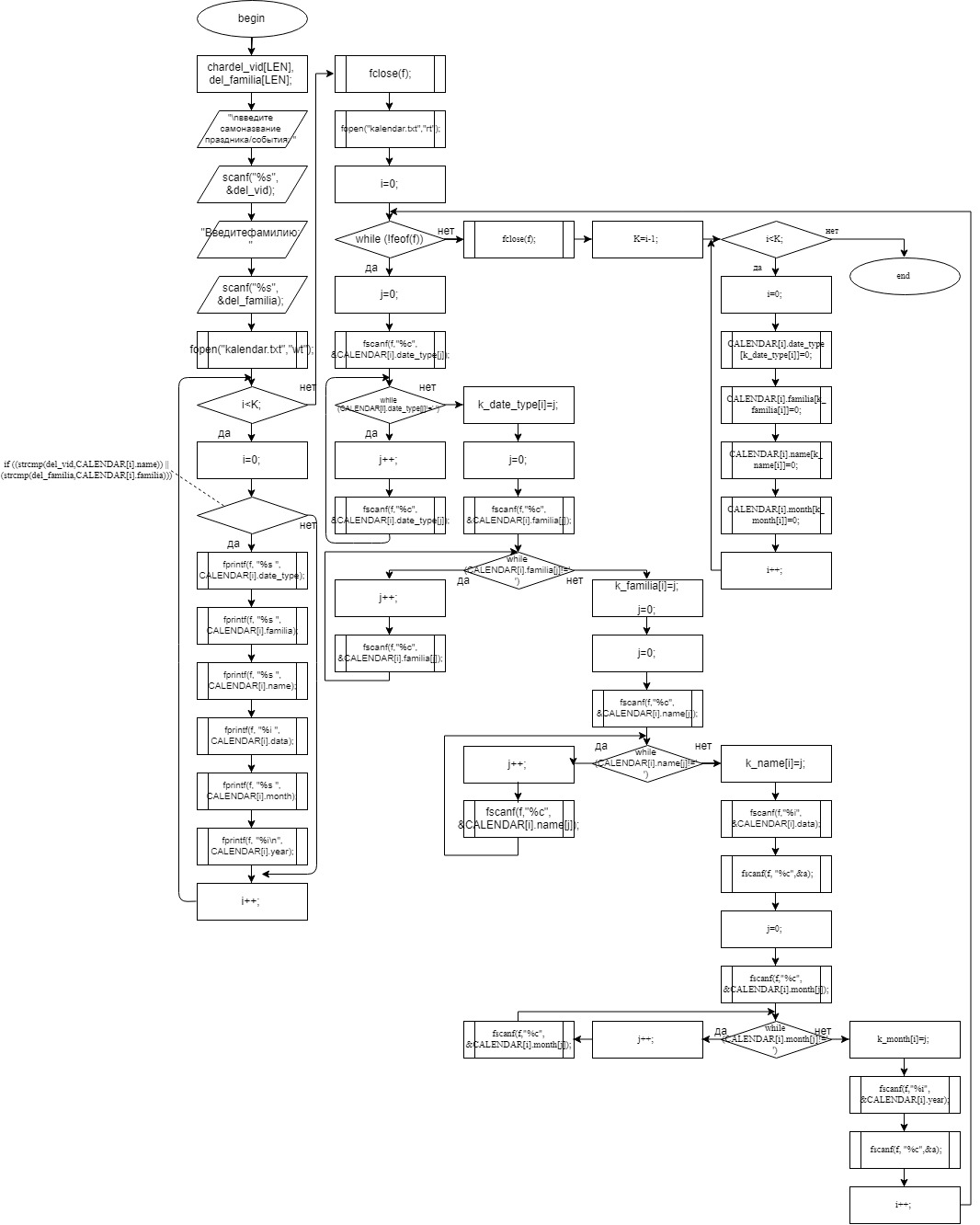


Рис.4. Схема алгоритма участка основной программы «case 2».

3.2.5. Схема алгоритма участка основной программы «case 3».

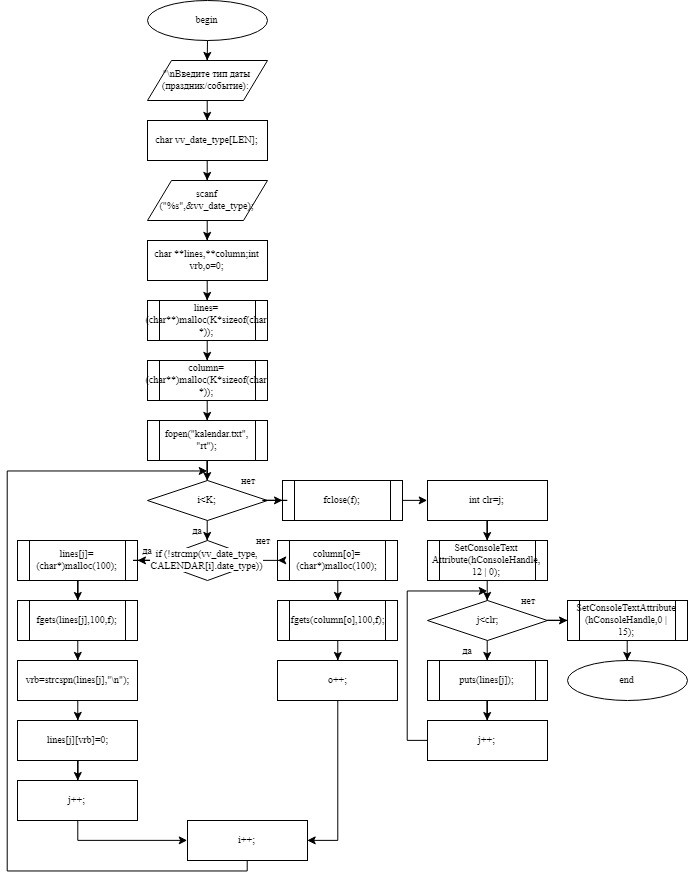


Рис.5. Схема алгоритма участка основной программы «case 3».

3.2.6. Схема алгоритма участка основной программы «case 4».

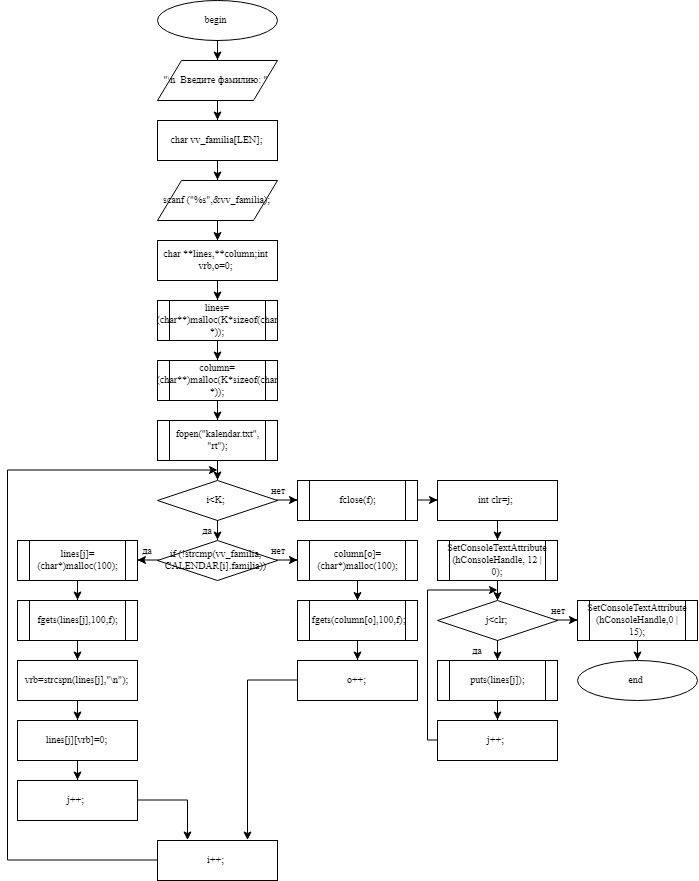


Рис.6. Схема алгоритма участка основной программы «case 4».

3.2.7. Схема алгоритма участка основной программы «case 5».

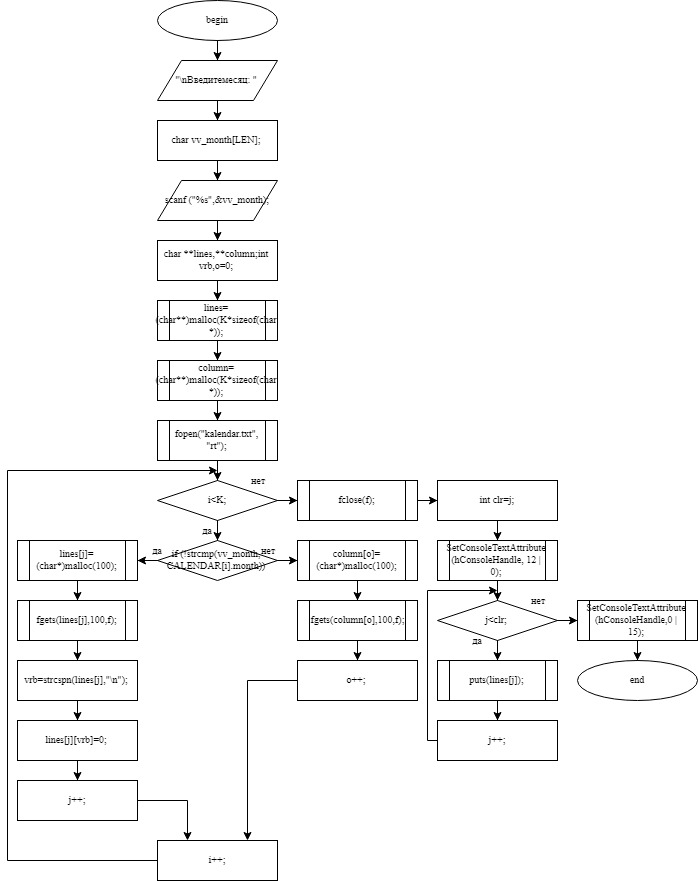


Рис.7. Схема алгоритма участка основной программы «case 5».

3.2.8 Схема алгоритма основной программы.

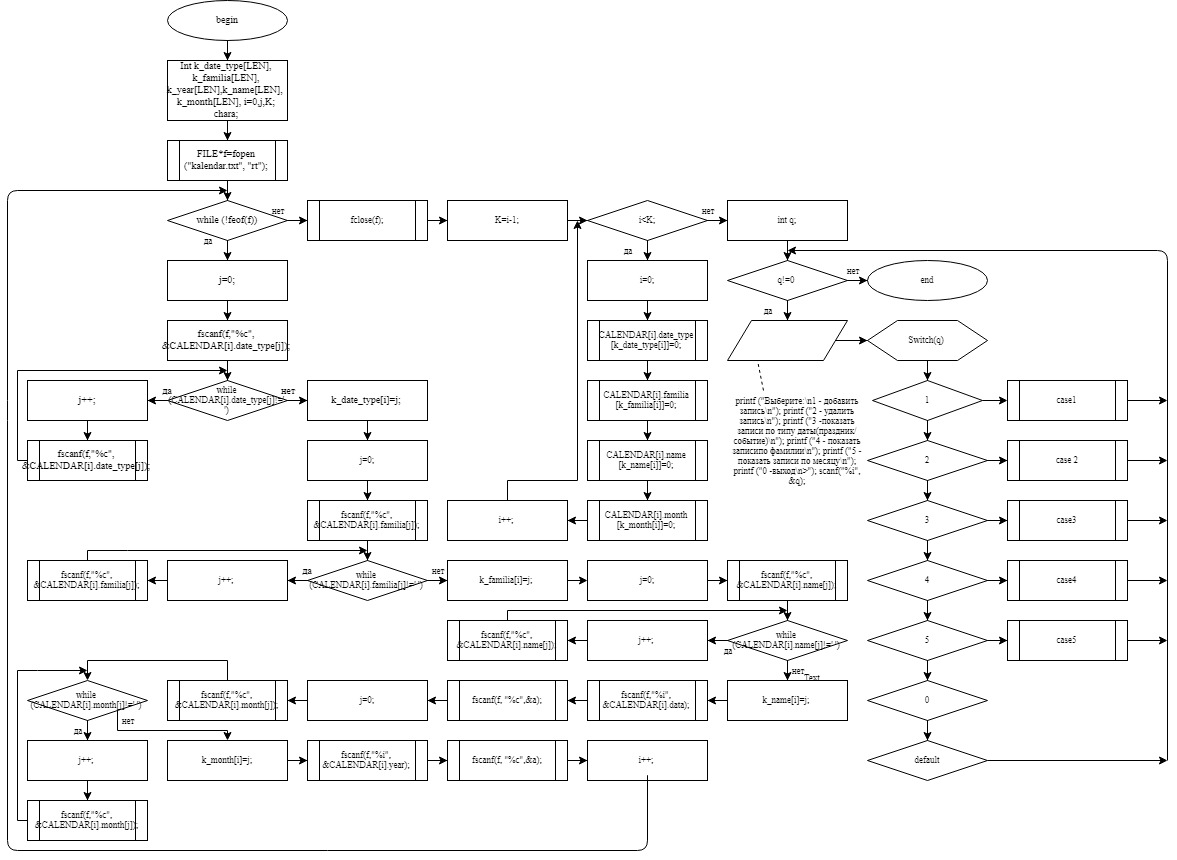


Рис.8. Схема алгоритма основной программы.

1. ТЕКСТ ПРОГРАММЫ

#include <stdio.h>

#include <windows.h>

#include <locale.h>

#define LEN 50

struct CALENDAR

{

char date\_type[50];

char familia[50];

char name[50];

int data;

char month[50];

int year;

}CALENDAR[LEN];

int compereString (const char \*\*x, const char \*\*y)

{

int rez=strcmp(\*x,\*y);

return rez;

}

func\_dobvleniya\_dat ()

{

int a;

char b[100];

FILE \*f=fopen ("kalendar.txt", "at");

printf ("\nВведите тип даты(праздник/событие): ");

scanf ("%s", &b);

fprintf (f, "%s ", b);

printf ("Введите фамилию: ");

scanf ("%s", &b);

fprintf (f, "%s ", b);

printf ("Введите само название праздника/события: ");

scanf ("%s", &b);

fprintf (f, "%s ", b);

printf ("Введите день: ");

scanf ("%i", &a);

fprintf (f, "%i ", a);

printf ("Введите месяц: ");

scanf ("%s", &b);

fprintf (f, "%s ", b);

printf ("Введите год: ");

scanf ("%i", &a);

fprintf (f, "%i\n", a);

fclose(f);

}

main ()

{

HANDLE hConsoleHandle=GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsoleHandle, 0 | 15);

setlocale(LC\_ALL, "Rus");

int k\_date\_type[LEN],k\_familia[LEN],k\_year[LEN], k\_name[LEN], k\_month[LEN];

int i=0,j,K;

char a;

FILE \*f=fopen("kalendar.txt", "rt");

while (!feof(f))

{

j=0;

fscanf(f, "%c", &CALENDAR[i].date\_type[j]);

while (CALENDAR[i].date\_type[j]!=' ')

{

j++;

fscanf(f, "%c", &CALENDAR[i].date\_type[j]);

}

k\_date\_type[i]=j;

j=0;

fscanf(f, "%c", &CALENDAR[i].familia[j]);

while (CALENDAR[i].familia[j]!=' ')

{

j++;

fscanf(f, "%c", &CALENDAR[i].familia[j]);

}

k\_familia[i]=j;

j=0;

fscanf(f, "%c", &CALENDAR[i].name[j]);

while (CALENDAR[i].name[j]!=' ')

{

j++;

fscanf(f, "%c", &CALENDAR[i].name[j]);

}

k\_name[i]=j;

fscanf(f, "%i", &CALENDAR[i].data);

fscanf(f, "%c", &a);

j=0;

fscanf(f, "%c", &CALENDAR[i].month[j]);

while (CALENDAR[i].month[j]!=' ')

{

j++;

fscanf(f, "%c", &CALENDAR[i].month[j]);

}

k\_month[i]=j;

fscanf(f, "%i", &CALENDAR[i].year);

fscanf(f, "%c", &a);

i++;

}

fclose(f);

K=i-1;

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

{

CALENDAR[i].date\_type[k\_date\_type[i]]=0;

CALENDAR[i].familia[k\_familia[i]]=0;

CALENDAR[i].name[k\_name[i]]=0;

CALENDAR[i].month[k\_month[i]]=0;

}

int q;

while (q!=0)

{

printf ("Выберите:\n1 - добавить запись\n");

printf ("2 - удалить запись\n");

printf ("3 - показать записи по типу даты(праздник/событие)\n");

printf ("4 - показать записи по фамилии\n");

printf ("5 - показать записи по месяцу\n");

printf ("0 - выход\n>");

scanf ("%i", &q);

switch (q)

{

case 1:

{

func\_dobvleniya\_dat();

FILE \*f=fopen("kalendar.txt", "rt");

i=0;

while (!feof(f))

{

j=0;

fscanf(f, "%c", &CALENDAR[i].date\_type[j]);

while (CALENDAR[i].date\_type[j]!=' ')

{

j++;

fscanf(f, "%c", &CALENDAR[i].date\_type[j]);

}

k\_date\_type[i]=j;

j=0;

fscanf(f, "%c", &CALENDAR[i].familia[j]);

while (CALENDAR[i].familia[j]!=' ')

{

j++;

fscanf(f, "%c", &CALENDAR[i].familia[j]);

}

k\_familia[i]=j;

j=0;

fscanf(f, "%c", &CALENDAR[i].name[j]);

while (CALENDAR[i].name[j]!=' ')

{

j++;

fscanf(f, "%c", &CALENDAR[i].name[j]);

}

k\_name[i]=j;

fscanf(f, "%i", &CALENDAR[i].data);

fscanf(f, "%c", &a);

j=0;

fscanf(f, "%c", &CALENDAR[i].month[j]);

while (CALENDAR[i].month[j]!=' ')

{

j++;

fscanf(f, "%c", &CALENDAR[i].month[j]);

}

k\_month[i]=j;

fscanf(f, "%i", &CALENDAR[i].year);

fscanf(f, "%c", &a);

i++;

}

fclose(f);

K=i-1;

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

{

CALENDAR[i].date\_type[k\_date\_type[i]]=0;

CALENDAR[i].familia[k\_familia[i]]=0;

CALENDAR[i].name[k\_name[i]]=0;

CALENDAR[i].month[k\_month[i]]=0;

}

break;

}

case 2:

{

char del\_vid[LEN], del\_familia[LEN];

printf ("\nвведите само название праздника/события: ");

scanf ("%s", &del\_vid);

printf ("Введите фамилию: ");

scanf ("%s", &del\_familia);

fopen("kalendar.txt", "wt");

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

{

if ((strcmp(del\_vid,CALENDAR[i].name)) || (strcmp(del\_familia,CALENDAR[i].familia)))

{

fprintf (f, "%s ", CALENDAR[i].date\_type);

fprintf (f, "%s ", CALENDAR[i].familia);

fprintf (f, "%s ", CALENDAR[i].name);

fprintf (f, "%i ", CALENDAR[i].data);

fprintf (f, "%s ", CALENDAR[i].month);

fprintf (f, "%i\n", CALENDAR[i].year);

}

}

fclose(f);

fopen("kalendar.txt", "rt");

i=0;

while (!feof(f))

{

j=0;

fscanf(f, "%c", &CALENDAR[i].date\_type[j]);

while (CALENDAR[i].date\_type[j]!=' ')

{

j++;

fscanf(f, "%c", &CALENDAR[i].date\_type[j]);

}

k\_date\_type[i]=j;

j=0;

fscanf(f, "%c", &CALENDAR[i].familia[j]);

while (CALENDAR[i].familia[j]!=' ')

{

j++;

fscanf(f, "%c", &CALENDAR[i].familia[j]);

}

k\_familia[i]=j;

j=0;

fscanf(f, "%c", &CALENDAR[i].name[j]);

while (CALENDAR[i].name[j]!=' ')

{

j++;

fscanf(f, "%c", &CALENDAR[i].name[j]);

}

k\_name[i]=j;

fscanf(f, "%i", &CALENDAR[i].data);

fscanf(f, "%c", &a);

j=0;

fscanf(f, "%c", &CALENDAR[i].month[j]);

while (CALENDAR[i].month[j]!=' ')

{

j++;

fscanf(f, "%c", &CALENDAR[i].month[j]);

}

k\_month[i]=j;

fscanf(f, "%i", &CALENDAR[i].year);

fscanf(f, "%c", &a);

i++;

}

fclose(f);

K=i-1;

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

{

CALENDAR[i].date\_type[k\_date\_type[i]]=0;

CALENDAR[i].familia[k\_familia[i]]=0;

CALENDAR[i].name[k\_name[i]]=0;

CALENDAR[i].month[k\_month[i]]=0;

}

break;

}

case 3:

{

printf ("\nВведите тип даты (праздник/событие): ");

char vv\_date\_type[LEN];

scanf ("%s", &vv\_date\_type);

char \*\*lines,\*\*column;

int vrb,o=0;

lines=(char \*\*)malloc(K\*sizeof(char \*));

column=(char \*\*)malloc(K\*sizeof(char \*));

j=0;

fopen("kalendar.txt", "rt");

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

{

if (!strcmp(vv\_date\_type,CALENDAR[i].date\_type))

{

lines[j]=(char\*)malloc(100);

fgets(lines[j],100,f);

vrb=strcspn(lines[j], "\n");

lines[j][vrb]=0;

j++;

}

else

{

column[o]=(char\*)malloc(100);

fgets(column[o],100,f);

o++;

}

}

fclose(f);

int clr=j;

SetConsoleTextAttribute(hConsoleHandle, 12 | 0);

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

{

puts(lines[j]);

}

SetConsoleTextAttribute(hConsoleHandle, 0 | 15);

break;

}

case 4:

{

printf ("\nВведите фамилию: ");

char vv\_familia[LEN];

scanf ("%s", &vv\_familia);

char \*\*lines,\*\*column;

int vrb,o=0;

lines=(char \*\*)malloc(K\*sizeof(char \*));

column=(char \*\*)malloc(K\*sizeof(char \*));

j=0;

fopen("kalendar.txt", "rt");

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

{

if (!strcmp(vv\_familia,CALENDAR[i].familia))

{

lines[j]=(char\*)malloc(100);

fgets(lines[j],100,f);

vrb=strcspn(lines[j], "\n");

lines[j][vrb]=0;

j++;

}

else

{

column[o]=(char\*)malloc(100);

fgets(column[o],100,f);

o++;

}

}

fclose(f);

int clr=j;

SetConsoleTextAttribute(hConsoleHandle, 12 | 0);

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

{

puts(lines[j]);

}

SetConsoleTextAttribute(hConsoleHandle, 0 | 15);

break;

}

case 5:

{

printf ("\nВведите месяц: ");

char vv\_month[LEN];

scanf ("%s", &vv\_month);

char \*\*lines,\*\*column;

int vrb,o=0;

lines=(char \*\*)malloc(K\*sizeof(char \*));

column=(char \*\*)malloc(K\*sizeof(char \*));

j=0;

fopen("kalendar.txt", "rt");

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

{

if (!strcmp(vv\_month,CALENDAR[i].month))

{

lines[j]=(char\*)malloc(100);

fgets(lines[j],100,f);

vrb=strcspn(lines[j], "\n");

lines[j][vrb]=0;

j++;

}

else

{

column[o]=(char\*)malloc(100);

fgets(column[o],100,f);

o++;

}

}

fclose(f);

int clr=j;

SetConsoleTextAttribute(hConsoleHandle, 12 | 0);

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

{

puts(lines[j]);

}

SetConsoleTextAttribute(hConsoleHandle, 0 | 15);

break;

}

case 0: break;

default: break;

}

printf ("\n");

}

}

1. ИНСТРУКЦИЯ ПО ПРИМЕНЕНИЮ

Данный программный продукт предназначен для пополнения базы данных «Календарь» с указанием типа даты (праздник/событие), фамилии, самого названия даты (праздника/события), числа, месяца и года, а также удаления данных из базы. Кроме этого программа производит вывод записей по категориям (праздник/событие), фамилиям и месяцам.

Для выбора действия после запуска программы, необходимо ввести номер в соответствии с инструкцией, появившейся на экране:

1 – внесение записи в базу. Для правильной работы необходимо следовать указаниям на экране.

2 – удаление записи из базы. Для правильной работы необходимо следовать указаниям на экране.

3 - вывод содержимого базы по категориям (праздник/событие).

4 - вывод записей по фамилии.

5 – вывод записей на указанный месяц.

0 - выход.

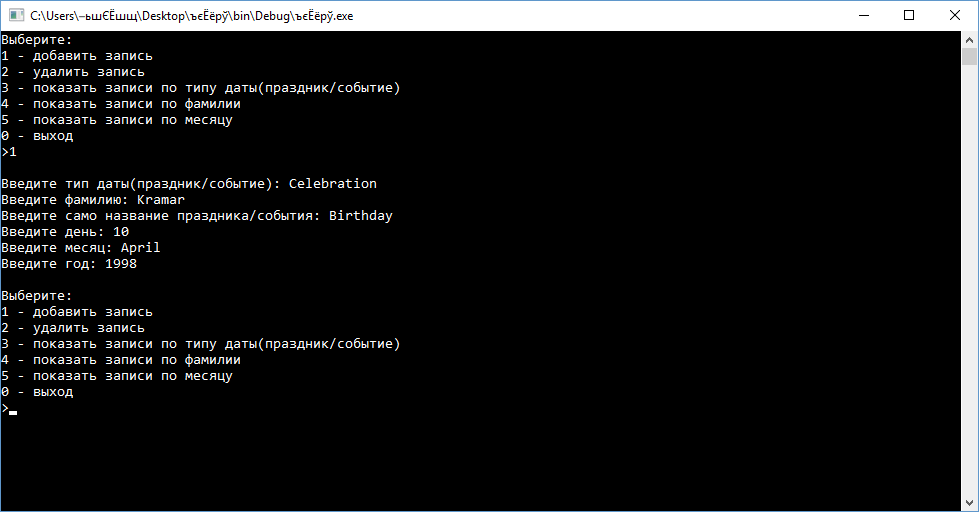


Рис. 1. Внесение записей в базе

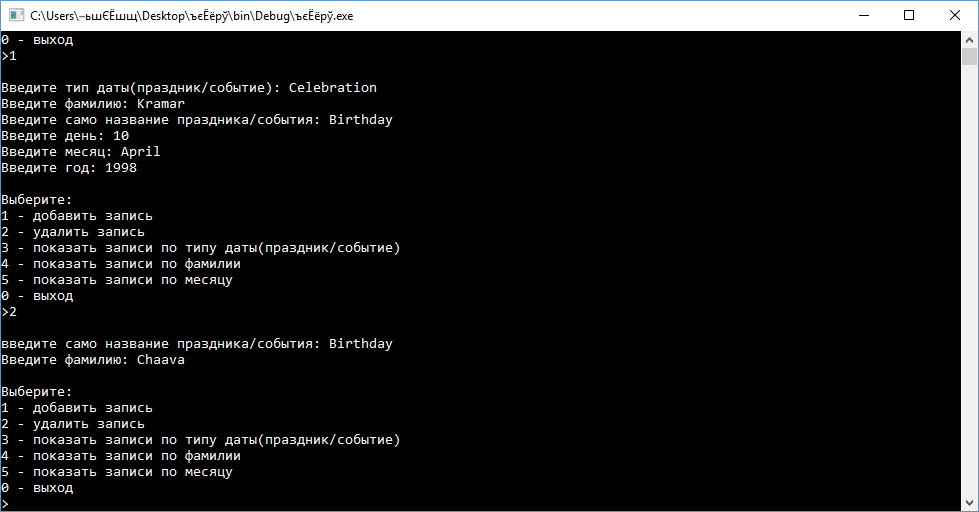


Рис.2. Удаление записей из базы

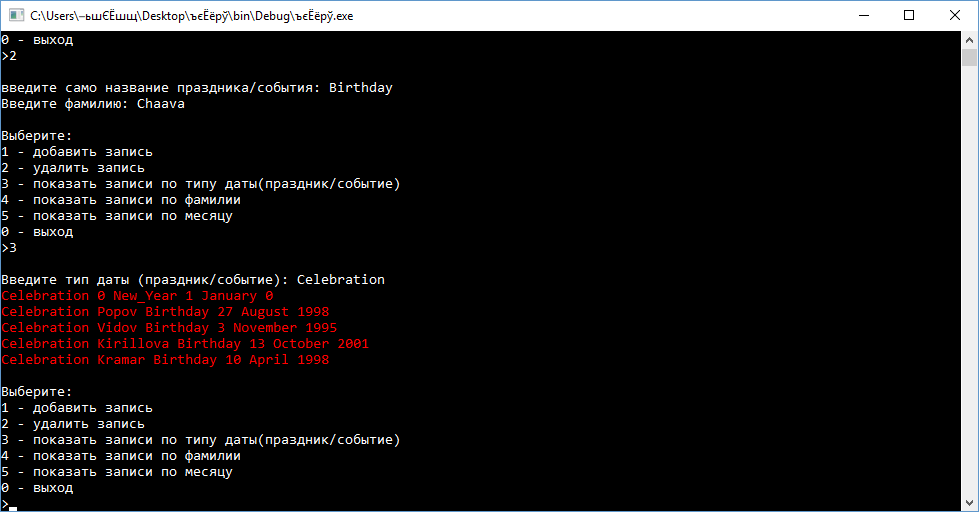


Рис. 3. Вывод записей по типу даты (праздник/событие)

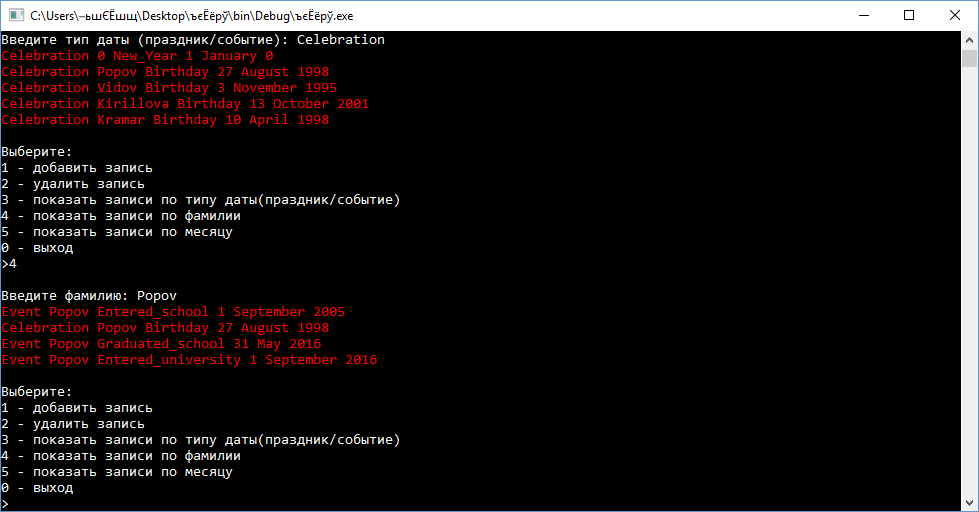
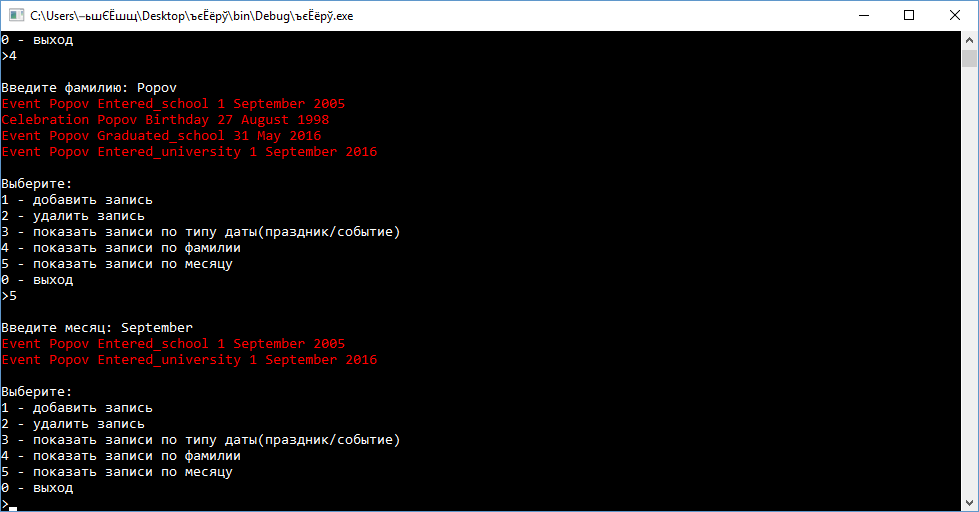


Рис. 4. Вывод записей по фамилии

  
Рис. 5. Вывод записей по месяцу

1. СПЕЦИФИКАЦИЯ ПРОГРАММЫ

|  |  |
| --- | --- |
| Calendar.c | Исполняемый модуль |
| stdio.h | Библиотека |
| locale.h | Библиотека |
| windows.h | Библиотека |
| kalendar.txt | Текстовый файл с данными |

ЗАКЛЮЧЕНИЕ

Работоспособность программы подтверждена примерами в пункте 5 данного отчёта. В программе выполняется пополнение базы данных «Календарь» с клавиатуры, производится удаление данных из базы, осуществляется вывод на экран данных из базы по категориям (праздник/событие), фамилиям и месяцам. Перезапись базы данных производится только после пополнения и удаления данных. Библиотека locale.h позволяет использовать кириллицу для лучшего восприятия пользователем.

СПИСОК ЛИТЕРАТУРЫ

1. Брайан Керниган, Деннис Ритчи – Язык программирования Си. – 3-е издание. — 2001г. – 142 с.
2. Семакин И. Г., Шестаков А. П. – Основы программирования – Москва: Издательство центр «Академия», 2004г. – 257 с.
3. Единая система программной документации. Гост 19.701-78.