

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра Электронных приборов (ЭП)

БАЗА ДАННЫХ МУЗЫКАЛЬНЫХ ПРОИЗВЕДЕНИЙ

Курсовой проект
по дисциплине «Информатика»

Пояснительная записка

Студент гр. 341:

_____ А.М. Зильберман

«__» _____ 2022 г.

Проверил:

ассистент кафедры ЭП

_____ А.А. Колмаков

оценка

«__» _____ 2022 г.

Томск 2022

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра электронных приборов (ЭП)

УТВЕРЖДАЮ

Заведующий кафедрой ЭП

д. ф.-м. н, профессор

_____ Н.И. Буримов

«___» _____ 2022г.

ЗАДАНИЕ

На курсовой проект Зильберману Артуру Максимовичу группы 341
факультета электронной техники.

1. Тема работы: «База данных».

2. Перечень подлежащих разработке вопросов:

2.1. Написание программы, которая создаёт базу данных, хранящую
информацию на жёстком диске.

3. Дата выдачи задания: «___» _____ 2022 г.

4. Срок сдачи законченной работы «___» _____ 2022 г.

Проверил:

ассистент кафедры ЭП Александр Андреевич Колмаков

«___» _____ 2022 г. _____

Задание принял к исполнению:

студент группы 341 Зильберман Артур Максимович

«___» _____ 2022 г. _____

Оглавление

1. ВВЕДЕНИЕ.....	4
1.1 Целесообразность проекта.....	4
1.2 Постановка задачи.....	4
1.3 Обзор предметной части.....	4
2. ОСНОВНАЯ ЧАСТЬ.....	6
2.1 Теоретическая часть.....	6
2.2 Проектное решение.....	8
2.3 Техническая реализация.....	8
2.4 Результаты работы.....	19
3. ЗАКЛЮЧЕНИЕ.....	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	23
Приложение А.....	24

1. ВВЕДЕНИЕ

1.1 Целесообразность проекта

В любое время людям было необходимо как-либо сохранять свои знания, поскольку человек не мог запомнить всё. Для этого изобретались различные способы, будь то наскальные рисунки или записи на бумаге. В всех современных организациях используются компьютеры для обработки и хранения служебной информации. Особую роль в этом играют базы данных, которые помогают автоматизировать и сортировать нужные сведения [1].

Целью данного курсового проекта является создание программы, которая предназначена для хранения информации о музыкальных произведениях, а также поиска объектов по заданным параметрам и их сортировки.

1.2 Постановка задачи

В соответствии с поставленной целью программа должна решать следующие задачи:

- 1) долговременное хранение информации об объектах в базе данных на жестком диске;
- 2) просмотр списка объектов;
- 3) сортировка списка по разным критериям;
- 4) добавление информации о новых объектах;
- 5) изменение информации об объектах и сохранение изменений в базе данных;
- 6) удаление информации об объектах из базы данных.

1.3 Обзор предметной области

В курсовом проекте отображена база музыкальных произведений. Создан простой интерфейс, который содержит основные функции для работы с базой данных:

1. Создание базы данных;
2. Ввод данных;
3. Просмотр;
4. Удаление;
5. Сортировка;
6. Поиск;
7. Количество элементов;
8. Выход.

2. ОСНОВНАЯ ЧАСТЬ

2.1 Теоретическая часть

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных, или, для краткости, просто базой данных.

Базы данных значительно изменились с момента их появления в начале 1960-х годов. Исходными системами, которые использовались для хранения и обработки данных, были навигационные базы данных — например, иерархические базы данных (которые опирались на древовидную модель и допускали только отношение «один-ко-многим») и базы данных с сетевой структурой (более гибкая модель, допускающая множественные отношения). Несмотря на простоту, эти ранние системы были негибкими. В 1980-х годах стали популярными реляционные базы данных, в 1990-х годах за ними последовали объектно-ориентированные базы данных. Совсем недавно вследствие роста Интернета и возникновения необходимости анализа неструктурированных данных появились базы данных NoSQL. В настоящее время облачные базы данных и автономные базы данных открывают новые возможности в отношении способов сбора, хранения, использования данных и управления ими.

Типы баз данных:

1. Реляционные базы данных. Они стали преобладать в 1980-х годах. Данные в реляционной базе организованы в виде таблиц, состоящих из столбцов и строк. Реляционная СУБД обеспечивает быстрый и эффективный доступ к структурированной информации.

2. Объектно-ориентированные базы данных. Информация в объектно-ориентированной базе данных представлена в форме объекта, как в объектно-ориентированном программировании.

3. Распределенные базы данных. Распределенная база данных состоит из двух или более частей, расположенных на разных серверах. Такая база данных может храниться на нескольких компьютерах.

4. Хранилища данных. Будучи централизованным репозиторием для данных, хранилище данных представляет собой тип базы данных, специально предназначенной для быстрого выполнения запросов и анализа.

5. Облачные базы данных. Облачная база данных представляет собой набор структурированных или неструктурированных данных, размещенный на частной, публичной или гибридной платформе облачных вычислений. Существует два типа моделей облачных баз данных: традиционная база данных и база данных как услуга (DBaaS). В модели DBaaS административные задачи и обслуживание выполняются поставщиком облачных услуг.

Подробнее о них можно прочитать в учебном пособии для вузов по информатике по базовому курсу, автор которого С.В. Симонович [1].

При выполнении курсового проекта мы будем использовать язык программирования Pascal. Pascal — один из наиболее известных языков программирования, используется для обучения программированию в старших классах и на первых курсах вузов, является основой для ряда других языков. Особенности языка являются строгая типизация и наличие средств структурного (процедурного) программирования. Наряду со строгой типизацией, в Паскале сведены к минимуму возможные синтаксические неоднозначности, а сам синтаксис автор постарался сделать интуитивно понятным даже при первом знакомстве с языком. [2].

При составлении отчёта по курсовому проекту, мы руководствовались методичкой «Методические указания к лабораторным работам» за авторством Е.С. Шандарова [3].

2.2 Проектное решение

На данном этапе формирования программы определены цель, задачи, необходимый функционал.

База данных будет содержать следующие данные:

- Название музыкальной группы
- Название музыкального произведения
- Родина музыкальной группы
- Длина музыкального произведения (в секундах)

Для написания программы базы данных был выбран язык программирования PASCAL.

Для организации эффективной работы с экраном, клавиатурой и встроенным динамиком используется модуль Crt. В данной программе также используется подпрограмма модуля ClrScr, которая очищает окно и помещает курсор в его верхний левый угол, а все позиции символов заполняются пробелами.

2.3 Техническая реализация

2.3.1 Обозначение типов данных

В данном разделе программы мы объявляем основные типы данных и подключаем модуль Crt.

```
program EPKP;  
uses crt;  
  
type fonoteka = record  
    MusicBand: string [30];  
    MusicName: string [30];  
    Lang: string[15];  
    SongDuration: integer;  
    Genre: string [15];  
    Year: integer;
```


end;

2.3.2 Процедура создания базы данных

На данном этапе происходит создание базы данных. Оператор ClrScr производит очистку экрана и перемещает курсор в его верхний левый угол. Оператор assign помогает связать запись с самим файлом. Оператор rewrite создаёт или, удаляя уже созданный, пересоздаёт пустой файл. Оператор close закрывает файл.

```
procedure CreateFile();
  var
    f: file of fonoteka;
  begin
    clrscr;
    Assign(f, 'Fonoteka.txt');
    Rewrite(f);
    Close(f);
  end;
```

2.3.3 Процедура добавления элемента

В данном блоке программы мы используем процедуру reset для открытия файла для чтения, функцию filesize для перемещения в переменную size количество записей, обозначая размер файла, и seek для перевода курсора в конец файла. Процедура write перемещает данные, добавленные в запись, в файл.

```
procedure AddElem();
  var
    f: file of fonoteka;
    a: fonoteka;
    size: integer;
```

```
b: integer;  
  
begin  
    clrscr;  
    writeln('Введите необходимые данные');  
    write('Название исполнителя/группы: ');  
    readln(a.MusicBand);  
    write('Название произведения: ');  
    readln(a.MusicName);  
    write('Язык исполнения: ');  
    readln(a.Lang);  
    write('Продолжительность произведения (в секундах):  
' );  
    readln(a.SongDuration);  
    write('Жанр: ');  
    readln(a.Genre);  
    write('Год выпуска произведения: ');  
    readln(a.Year);  
    Assign(f, 'Fonoteka.txt');  
    reset(f);  
    size := FileSize(f);  
    seek(f, size);  
    write(f, a);  
    close(f);  
    clrscr;  
  
end;
```

2.3.4. Процедура просмотра базы данных

Процедура позволяет посмотреть данные из всех добавленных записей. Цикл «while not eof(f) do begin [оператор] end» обеспечивает работу оператора, пока не закончится файл f. Пока процедура идёт до конца файла, на экран выводятся введённые ранее записи, которым присваивается для удобства условный номер i, который в памяти не сохраняется.

```
procedure CheckBaza ();
var
  f:file of fonoteka;
  r:fonoteka;
  i:integer;
begin
  clrscr;
  assign(f, 'Fonoteka.txt');
  reset(f);
  while not eof(f) do
    begin
      read(f, r);
      i := i + 1;
      Writeln('№ ', i);
      Writeln ('Название исполнителя/группы: ',
r.MusicBand);
      Writeln('Название произведения: ', r.MusicName);
      Writeln('Язык исполнения: ', r.Lang);
      Writeln('Продолжительность произведения: ',
r.SongDuration , ' сек');
      Writeln('Жанр: ', r.Genre);
      Writeln('Год выпуска: ', r.Year);
      Writeln();
```

```

end;

readln;

close(f);

end;

```

2.3.5 Процедура удаления элемента

Данная процедура даёт возможность удалить запись из файла. В ней используется процедура `rename`, которая переименовывает файл, а также `erase`, которая удаляет файл. Помимо этого реализована процедура подтверждения удаления. В ходе процедуры все записи, кроме выбранной, перемещаются в новый файл “Delete.txt”, который в конце процедуры заменяет собой “Fonoteka.txt”.

```

procedure DeleteElem();
  var
    f, g: file of fonoteka;
    a: fonoteka;
    n, i: integer;
    answer: string;
  begin
    clrscr;
    write('Введите номер удаляемой записи:');
    readln(n);
    writeln ('Вы уверены, что хотите удалить ', n, '
элемент из базы данных? (да/нет)');
    repeat
      readln(answer);
      if not (answer = 'да') and not (answer = 'нет')
      then writeln('Введите отмет корректно');
    until (answer = 'да') or (answer = 'нет');
  end;

```

```

if (answer = 'да') then
    begin
        assign(f, 'Fonoteka.txt');
        assign(g, 'Delete.txt');
        reset(f);
        rewrite(g);
        while not eof(f) do
            begin
                read(f, a);
                i := i + 1;
                if i <> n then write(g, a);
            end;
        close(f);
        close(g);
        erase(f);
        rename(g, 'Fonoteka.txt');
        writeln('Запись удалена');
        readln;
    end
else if (answer = 'нет') then
    begin
        writeln('Удаление отменено');
    end
end;

```

2.3.6 Процедура сортировки

Данный блок программы реализует сортировку записей с помощью массивов методом пузырька, суть которого заключается в многократном прохождении по списку, на каждом шаге сравнивая пары элементов, и если

порядок в паре неверный, то элементы в паре меняются местами. Таким образом все записи списка выстаиваются в нужном порядке. Ниже представлен пример сортировки, использованный в программе.

```
procedure Sort_Year();
  var
    f: file of fonoteka;
    arr: array[1..100] of fonoteka;
    r: fonoteka;
    i, j, k: integer;
begin
  assign(f, 'Fonoteka.txt');
  reset(f);
  while not eof(f) do
    begin
      i := i + 1;
      read(f, arr[i]);
    end;
    for k := 1 to i - 1 do
      for j := 1 to i - k do
        begin
          if (arr[j].Year > arr[j + 1].Year) then
            begin
              r := arr[j];
              arr[j] := arr[j + 1];
              arr[j + 1] := r;
            end;
          end;
        end;
      clrscr;
    for k := 1 to i do
```

```

begin

    Write('Название исполнителя/группы: ');
    Writeln(arr[k].MusicBand);
    Write('Название произведения: ');
    Writeln(arr[k].MusicName);
    Write('Язык исполнения: ');
    Writeln(arr[k].Lang);
    Write('Продолжительность произведения: ');
    Writeln(arr[k].SongDuration, 'сек');
    Write('Жанр: ');
    Writeln(arr[k].Genre);
    Write('Год выпуска: ');
    Writeln(arr[k].Year);
    Writeln;

end;

close(f);

readln;

end;

```

2.3.7 Процедура изменения информации в записи

На данном этапе при помощи процедуры мы меняем информацию в выбранной записи из базы данных. Процедура немного отличается от процедуры добавления элемента, перед этим лишь задаётся номер записи в базе, и затем он перезаписывается на новые внесённые данные. Так как подсчёт элементов с помощью команды seek начинается с нуля, то, для того чтобы изменить нужный элемент, нам нужно записать seek(f,n-1).

```

procedure ChangeElem();
var

```

```

f: file of fonoteka;
a: fonoteka;
size: integer;
n: integer;
answer: string;
begin
    clrscr;
    writeln('Укажите номер изменяемой записи');
    readln (n);
    writeln ('Вы уверены, что хотите изменить ', n, '
элемент из базы данных? (да/нет) ');
    repeat
        readln(answer);
        if not (answer = 'да') and not (answer = 'нет')
            then writeln('Введите ответ корректно');
    until (answer = 'да') or (answer = 'нет');
    if (answer = 'да') then
        begin
            writeln('Введите новые данные');
            write('Название исполнителя/группы: ');
            readln(a.MusicBand);
            write('Название произведения: ');
            readln(a.MusicName);
            write('Язык исполнения: ');
            readln(a.Lang);
            write('Продолжительность произведения (в
секундах): ');
            readln(a.SongDuration);
            write('Жанр: ');
            readln(a.Genre);

```



```

write('Год выпуска произведения: ');
readln(a.Year);
Assign(f, 'Fonoteka.txt');
reset(f);
seek(f, n-1);
write(f, a);
close(f);
clrscr;

end

else if (answer = 'нет') then
begin
    writeln('Изменение отменено');
end;
end;

```

2.3.8 Тело программы, блок меню

На этом этапе программы происходит выбор пункта меню и вызов соответствующей процедуры.

```

var
c:integer;
begin
    repeat
        clrscr;
        Writeln('Введите пункт меню... ');
        Writeln('1. Создать/пересоздать базу данных');
        Writeln('2. Посмотреть базу данных');
        Writeln('3. Сортировать по критерию...');
        Writeln('4. Добавить запись');
    
```

```
Writeln('5. Изменить запись');
Writeln('6. Удалить запись');
Writeln('7. Завершить сеанс');
Readln(c);

case c of
    1: CreateFile();
    2: CheckBaza();
    3: Sort();
    4: AddElem();
    5: ChangeElem();
    6: DeleteElem();
    7: exit;
end;

until c = 7;

end.
```

2.4. Результаты работы

При запуске программы появляется главное меню и пользователю предлагается выбрать действие, рис. 6.1:



Рис. 6.1. Меню

При выборе пункта 1, происходит создание базы данных. При выборе пункта 4, пользователю предлагается ввести данные к записи, рис. 6.2:



Рис. 6.2. Добавление новой записи в базу данных

При выборе пункта 2, пользователь может ознакомиться с содержимым базы данных, рис. 6.3:



Рис. 6.3. Просмотр базы данных

При выборе пункта 6, происходит удаление выбранного элемента рис. 6.4:

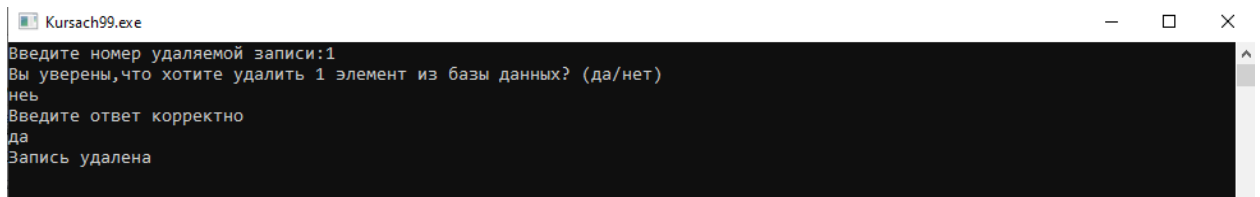


Рис. 6.4. Удаление элемента базы данных

При выборе пункта 3, появляется меню выбора критерия сортировки, рис 6.5, с последующей сортировкой по выбранному критерию, рис. 6.6:



Рис. 6.5. Меню выбора критерия сортировки

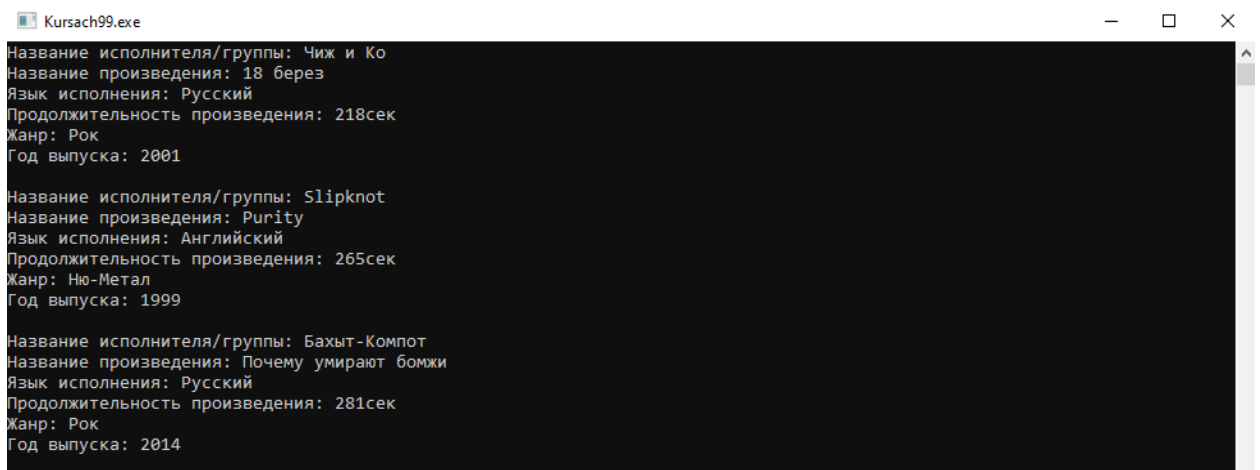


Рис. 6.6 Отсортированные данные

При выборе пункта 5, пользователю предлагается написать номер записи, который он хочет изменить, и после этого ввести новые данные, рис.6.7:

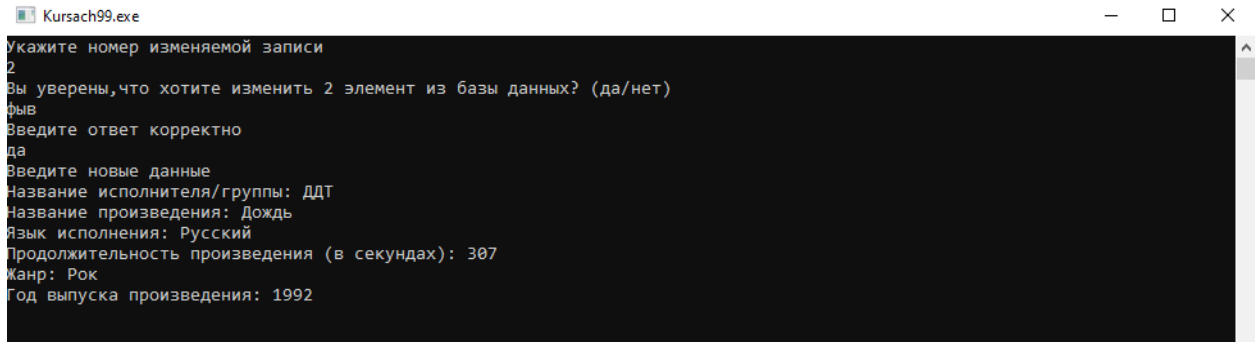


Рис. 6.7. Меню выбора нужной записи для редактирования и ввод новых данных

При выборе пункта 7, происходит завершение работы программы.

3. ЗАКЛЮЧЕНИЕ

В ходе курсовой работы были расширены и закреплены теоретические знания, а также были получены практические навыки путем решения задачи создания базы данных музыкальных произведений. Мы создали программу, способную сохранять введённую информацию о музыкальных произведениях в отдельном файле с расширением .txt. Таким образом, цель данной курсовой работы была достигнута, посредством решения поставленных задач.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Информатика. Базовый курс : учебное пособие для вузов / С. В. Симонович [и др.] ред. С. В. Симонович. - 2-е изд. - СПб. : Питер, 2008. - 639 с.
- [2] Кудинов Ю.И., Пащенко Ф.Ф. Основы современной информатики. 2-е изд. Испр. — СПб.: Издательство «Лань», 2011 г. — 256 с.
http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=2024
- [3] Шандаров Е.С. Методические указания к лабораторным работам, 2013.

Приложение А

```
program ЕПКР;  
uses crt;  
  
type fonoteka = record //присваиваем fonoteka тип  
record  
    MusicBand: string [30];  
    MusicName: string [30];  
    Lang: string[15];  
    SongDuration: integer;  
    Genre: string [15];  
    Year: integer;  
end;  
  
procedure CreateFile(); //процедура создания файла  
var  
    f: file of fonoteka; //файловая переменная  
begin  
    clrscr; //очистка экрана  
    Assign(f, 'Fonoteka.txt'); //связывание переменной с  
    файлом  
    Rewrite(f); //создание файла  
    Close(f); //закрытие файла  
end;  
  
procedure AddElem(); //процедура добавления элемента  
var  
    f: file of fonoteka;  
    a: fonoteka;  
    size: integer;
```



```

b: integer;

begin
    clrscr;
    writeln('Введите необходимые данные');
    write('Название исполнителя/группы: ');
    readln(a.MusicBand);
    write('Название произведения: ');
    readln(a.MusicName);
    write('Язык исполнения: ');
    readln(a.Lang);
    write('Продолжительность произведения (в секундах):
');
    readln(a.SongDuration);
    write('Жанр: ');
    readln(a.Genre);
    write('Год выпуска произведения: ');
    readln(a.Year);
    Assign(f, 'Fonoteka.txt');
    reset(f); //открытие файла на чтение
    size := FileSize(f); //размер файла
    seek(f, size); //установка указателя на конец файла
    write(f, a); //запись данных в файл
    close(f); //закрытие файла
    clrscr;

end;

procedure ChangeElem(); //процедура изменения записи
var
    f: file of fonoteka;

```

```

a: fonoteka;
size: integer;
n: integer;
answer: string;
begin
    clrscr;
    writeln('Укажите номер изменяемой записи');
    readln (n);
    writeln ('Вы уверены, что хотите изменить ', n, '
элемент из базы данных? (да/нет) ');
    repeat
        readln(answer);
        if not (answer = 'да') and not (answer = 'нет')
        then writeln('Введите ответ корректно');
    until (answer = 'да') or (answer = 'нет');
    if (answer = 'да') then
        begin
            writeln('Введите новые данные');
            write('Название исполнителя/группы: ');
            readln(a.MusicBand);
            write('Название произведения: ');
            readln(a.MusicName);
            write('Язык исполнения: ');
            readln(a.Lang);
            write('Продолжительность произведения (в
секундах): ');
            readln(a.SongDuration);
            write('Жанр: ');
            readln(a.Genre);
            write('Год выпуска произведения: ');

```

```

        readln(a.Year);
        Assign(f, 'Fonoteka.txt');
        reset(f);
        seek(f, n-1); //установка указателя на выбранный
элемент
        write(f, a);
        close(f);
        clrscr;
    end
    else if (answer = 'нет') then
    begin
        writeln('Изменение отменено');
    end;
end;

```

```

procedure CheckBaza(); //процедура просмотра данных
var
    f:file of fonoteka;
    r:fonoteka;
    i:integer;
begin
    clrscr;
    assign(f, 'Fonoteka.txt');
    reset(f);
    while not eof(f) do
    begin
        read(f, r);
        i := i + 1;
        Writeln('№ ', i);
    end;

```

```

        Writeln ('Название исполнителя/группы: ',
r.MusicBand);
        Writeln('Название произведения: ', r.MusicName);
        Writeln('Язык исполнения: ', r.Lang);
        Writeln('Продолжительность произведения: ',
r.SongDuration , ' сек');
        Writeln('Жанр: ', r.Genre);
        Writeln('Год выпуска: ', r.Year);
        Writeln();
    end;
    readln;
    close(f);
end;

```

procedure DeleteElem(); //процедура удаления записи

var

f, g: **file of** fonoteka;

a: fonoteka;

n, i: **integer**;

answer: **string**;

begin

clrscr;

write('Введите номер удаляемой записи:');

readln(n);

writeln ('Вы уверены, что хотите удалить ', n, ' элемент из базы данных? (да/нет)');

repeat

readln(answer);

if not (answer = 'да') **and not** (answer = 'нет')

then writeln('Введите ответ корректно');

```

until (answer = 'да') or (answer = 'нет');
if (answer = 'да') then
    begin
        assign(f, 'Fonoteka.txt');
        assign(g, 'Delete.txt');
        reset(f);
        rewrite(g);
        while not eof(f) do
            begin
                read(f, a);
                i := i + 1;
                if i <> n then write(g, a);
            end;
            close(f);
            close(g);
            erase(f);
            rename(g, 'Fonoteka.txt');
            writeln('Запись удалена');
            readln;
        end
    else if (answer = 'нет') then
        begin
            writeln('Удаление отменено');
        end
    end;

procedure Sort_Year(); //сортировка по году
var
    f: file of fonoteka;

```

```

arr: array[1..100] of fonoteka;
r: fonoteka;
i, j, k: integer;
begin
    assign(f, 'Fonoteka.txt');
    reset(f);
    while not eof(f) do
        begin
            i := i + 1;
            read(f, arr[i]);
        end;
        for k := 1 to i - 1 do
            for j := 1 to i - k do
                begin
                    if (arr[j].Year > arr[j + 1].Year) then
                        begin
                            r := arr[j];
                            arr[j] := arr[j + 1];
                            arr[j + 1] := r;
                        end;
                    end;
                end;
            end;
        end;
        clrscr;
        for k := 1 to i do
            begin
                Write('Название исполнителя/группы: ');
                Writeln(arr[k].MusicBand);
                Write('Название произведения: ');
                Writeln(arr[k].MusicName);
                Write('Язык исполнения: ');
            end;
        end;
    end;

```

```

        Writeln(arr[k].Lang);
        Write('Продолжительность произведения: ');
        Writeln(arr[k].SongDuration, 'сек');
        Write('Жанр: ');
        Writeln(arr[k].Genre);
        Write('Год выпуска: ');
        Writeln(arr[k].Year);
        Writeln;
    end;
    close(f);
    readln;
end;

```

procedure Sort_SongDuration(); //сортировка по
длительности песни

```

var
    f: file of fonoteka;
    arr: array[1..100] of fonoteka;
    r: fonoteka;
    i, j, k: integer;
begin
    assign(f, 'Fonoteka.txt');
    reset(f);
    while not eof(f) do
    begin
        i := i + 1;
        read(f, arr[i]);
    end;
    for k := 1 to i - 1 do

```

```

for j := 1 to i - k do
begin
    if (arr[j].SongDuration > arr[j +
1].SongDuration) then
        begin
            r := arr[j];
            arr[j] := arr[j + 1];
            arr[j + 1] := r;
        end;
    end;
end;

clrscr;

for k := 1 to i do
begin
    Write('Название исполнителя/группы: ');
    Writeln(arr[k].MusicBand);
    Write('Название произведения: ');
    Writeln(arr[k].MusicName);
    Write('Язык исполнения: ');
    Writeln(arr[k].Lang);
    Write('Продолжительность произведения: ');
    Writeln(arr[k].SongDuration, 'сек');
    Write('Жанр: ');
    Writeln(arr[k].Genre);
    Write('Год выпуска: ');
    Writeln(arr[k].Year);
    Writeln;
end;

close(f);

readln;

```



```

end;

procedure Sort(); //процедура вывода меню сортировки
var
h:integer;
begin
    repeat
        ClrScr();
        Writeln('Сортировать по:');
        Writeln('1. Названию исполнителя/группы');
        Writeln('2. Продолжительности произведения');
        Writeln('3. Выход из меню сортировки');
        readln(h);
        case h of //вызов процедуры указанной сортировки
            1: Sort_Year();
            2: Sort_SongDuration();
            3: begin
                    end;
                end;
        until h=3;
        Writeln();
    end;

var
c:integer;
begin
    repeat
        clrscr;
        Writeln('Введите пункт меню... ');

```

```
Writeln('1. Создать/пересоздать базу данных');
Writeln('2. Посмотреть базу данных');
Writeln('3. Сортировать по критерию...');
Writeln('4. Добавить запись');
Writeln('5. Изменить запись');
Writeln('6. Удалить запись');
Writeln('7. Завершить сеанс');
Readln(c); //ввод пункта меню

case c of //по выбранному пункту вызывается
указанная процедура
    1: CreateFile();
    2: CheckBaza();
    3: Sort();
    4: AddElem();
    5: ChangeElem();
    6: DeleteElem();
    7: exit;
end;
until c = 7;
end.
```