## Курсовая работа.

# *Программа управления базой данных.*

## Техническое задание.

База данных содержит одну таблицу, состоящую из записей по образцу, представленному в разделе «Варианты заданий». Номер варианта выбирается в соответствии с последней цифрой номера зачетной книжки, например, если шифр Пи-10-3814, то студент выбирает вариант №4 для исполнения. Запись – это совокупность данных, содержащихся в одной строке таблицы. При запуске программы все записи базы данных считываются из типизированного файла в массив данных того же типа и выводятся на экран построчно. В дальнейшем программа работает с элементами массива.

Требования к программе:

– база данных хранится в типизированном файле;
– доступ к данным осуществляется посредством меню пользователя;
– меню представлено совокупностью пунктов, выводимых на экран;
– по меню пользователь может перемещаться с помощью клавиш стрелок;
– текущее положение в меню должно выделяться цветом шрифта или фона;
– запуск действия по выбранному пункту меню осуществляется нажатием клавиши «Enter»;
– меню пользователя должно содержать обязательные пункты:

1) **добавить запись**
   – при выборе этого пункта пользователю предлагается последовательно ввести значения всех полей записи;
2) **удалить запись;**
3) **найти записи по критерию**
   – вывести все записи, где результат – последняя колонка таблицы – не хуже заданного пользователем, при выборе этого пункта программа сначала у пользователя спрашивает значение критерия – затем в ответ выдает список, например, вывести на экран всех студентов, у которых оценка не ниже «хорошо»;
4) **найти лучший результат**
   – вывести на экран запись/записи с лучшим результатом;
5) **сохранить изменения в файл;**
6) **выйти из программы**
   – программа должна спрашивать пользователя: «вы действительно хотите покинуть программу?».

## Оформление отчета о проделанной работе.

Требования к оформлению.

Объем отчета: 10-12 страниц (листы А4 формата, сброшюрованы):

1) Титульный лист,
2) Содержание,
3) Разработка программы.

   Описать логику работы программы (меню, чтение из файла, запись в файл, выборка из базы данных по критерию) с соответствующими выдержками программного кода, с ссылками на блок схему алгоритма (приложение №1) и, по необходимости, с примерами. Обосновать выбор методов обработки информации. По возможности привести части неоптимизированного программного кода, который получался в процессе разработки программы и указать найденные недостатки и порядок доработки алгоритма.

5) Приложение 1. Блок-схема алгоритма организации меню СУБД.
6) Приложение 2. Текст программы.

## Защита отчета о проделанной работе.

За две недели до сессии необходимо:

– прислать в секретариат заочного отделения (для кафедры ИС ФПИ ПГСХА) работу в бумажном варианте;

– на адрес algopro@narod.ru или иной, указанный преподавателем, прислать работу в электронном варианте;

– в электронном варианте должны быть – папка с файлами:

1) информация об авторе,

2) текст реферата,

3) все файлы программы (*.pas, *.tpu, *.exe, *.txt), обязательным условием является наличие откомпилированной программы в файл, исполняемый без среды программирования (*.exe)

Программы, содержащие плагиат, не допускаются к защите. Студенты, не приславшие в указанные сроки отчет, не допускаются к защите. Их защита переносится, а сроки обсуждаются индивидуально с преподавателем.

В установленное время студент прибывает на кафедру, имея с собой отчет о проделанной работе, который должен быть представлен:

– распечаткой текстов программ на листах формата А4;

– в электронном виде.

Порядок защиты отчета студентом:

– демонстрирует работоспособность разработанной программы;

– отвечает на вопросы преподавателя по тексту программного кода;

– выполняет дополнительное задание по модернизации программы.

*Критерии оценивания:*

«**отлично**» – программа работоспособна, студент свободно ориентируется в тексте программы и в состоянии его модернизировать по требованию преподавателя;

«**хорошо**» – программа работоспособна, студент ориентируется в тексте программы и в состоянии объяснить суть реализованных в программе алгоритмов;

«**удовлетворительно**» – некоторые возможности программы реализованы, студент в состоянии объяснить назначение отдельных блоков программы (процедур, функций), а также используемых структурных операторов;

«**неудовлетворительно**» – программа неработоспособна или студент не в состоянии объяснить назначение используемых структурных операторов.

## Варианты заданий:

1. База данных «Участники соревнований» – содержит таблицу, состоящую из записей:

| Имя поля | Фамилия | И | О | Возраст | Результат в баллах |
|---|---|---|---|---|---|
| Тип данных | String | Char | Char | Byte | Word |

Например:

| | | | | |
|---|---|---|---|---|
| Петров | В | А | 18 | 244 |
| Васечкин | П | О | 19 | 255 |
| Сидорова | М | В | 18 | 194 |

2. База данных «Участники соревнований» – содержит таблицу, состоящую из записей:

| Имя поля | Фамилия | И | О | Пол | Результат в сек |
|---|---|---|---|---|---|
| Тип данных | String | Char | Char | Char | Real |

Например:

| | | | | |
|---|---|---|---|---|
| Петров | В | А | м | 12.1 |
| Васечкин | П | О | м | 12.4 |
| Сидорова | М | В | ж | 13.4 |

3. База данных «Участники соревнований» – содержит таблицу, состоящую из записей:

| Имя поля | Фамилия | И | О | Знак Зодиака | Результат в баллах (от 1 до 10) |
|---|---|---|---|---|---|
| Тип данных | String | Char | Char | String | Byte |

Например:

| | | | | |
|---|---|---|---|---|
| Петров | В | А | Овен | 9 |
| Васечкин | П | О | Рак | 7 |
| Сидорова | М | В | Весы | 8 |

4. База данных «Участники соревнований» – содержит таблицу, состоящую из записей:

| Имя поля | Фамилия | И | О | Разряд | Результат (кол-во повторений) |
|---|---|---|---|---|---|
| Тип данных | String | Char | Char | String | Byte |

Например:

| | | | | |
|---|---|---|---|---|
| Петров | В | А | кмс | 24 |
| Васечкин | П | О | мс | 36 |
| Сидорова | М | В | 2 | 13 |

5. База данных «Участники соревнований» – содержит таблицу, состоящую из записей:

| Имя поля | Фамилия | И | О | Вес, кг | Результат в кг |
|---|---|---|---|---|---|
| Тип данных | String | Char | Char | Real | Real |

Например:

| | | | | |
|---|---|---|---|---|
| Петров | В | А | 68.75 | 120.5 |
| Васечкин | П | О | 67.50 | 130.0 |
| Сидорова | М | В | 54.75 | 98.5 |

6. База данных «Студенты» – содержит таблицу, состоящую из записей:

| Имя поля | Фамилия | И | О | Подгруппа | Результат экзамена |
|---|---|---|---|---|---|
| Тип данных | String | Char | Char | Char | Byte |

Например:

| | | | | |
|---|---|---|---|---|
| Петров | В | А | а | 5 |
| Васечкин | П | О | а | 4 |
| Сидорова | М | В | в | 4 |

7. База данных «Студенты» – содержит таблицу, состоящую из записей:

| Имя поля | Фамилия | И | О | Подгруппа | Результат (рейтинг) |
|---|---|---|---|---|---|
| Тип данных | String | Char | Char | Char | Real |

Например:

| | | | | |
|---|---|---|---|---|
| Петров | В | А | б | 95.5 |
| Васечкин | П | О | а | 104.5 |
| Сидорова | М | В | в | 84.0 |

8. База данных «Выпускники» – содержит таблицу, состоящую из записей:

| Имя поля | Фамилия | И | О | Специаль–ность | Результат (средн. оценка в дипломе) |
|---|---|---|---|---|---|
| Тип данных | String | Char | Char | String | Real |

Например:

| | | | | |
|---|---|---|---|---|
| Петров | В | А | инженер | 4.75 |
| Васечкин | П | О | инженер | 4.50 |
| Сидорова | М | В | инженер-экономист | 3.25 |

9. База данных «Выпускники» – содержит таблицу, состоящую из записей:

| Имя поля | Фамилия | И | О | Специаль–ность | Результат (оценка за диплом) |
|---|---|---|---|---|---|
| Тип данных | String | Char | Char | Word | Byte |

Например:

| | | | | |
|---|---|---|---|---|
| Петров | В | А | 080801 | 5 |
| Васечкин | П | О | 230201 | 4 |
| Сидорова | М | В | 230201 | 3 |

## Скриншоты примеров программы управления базой данных.

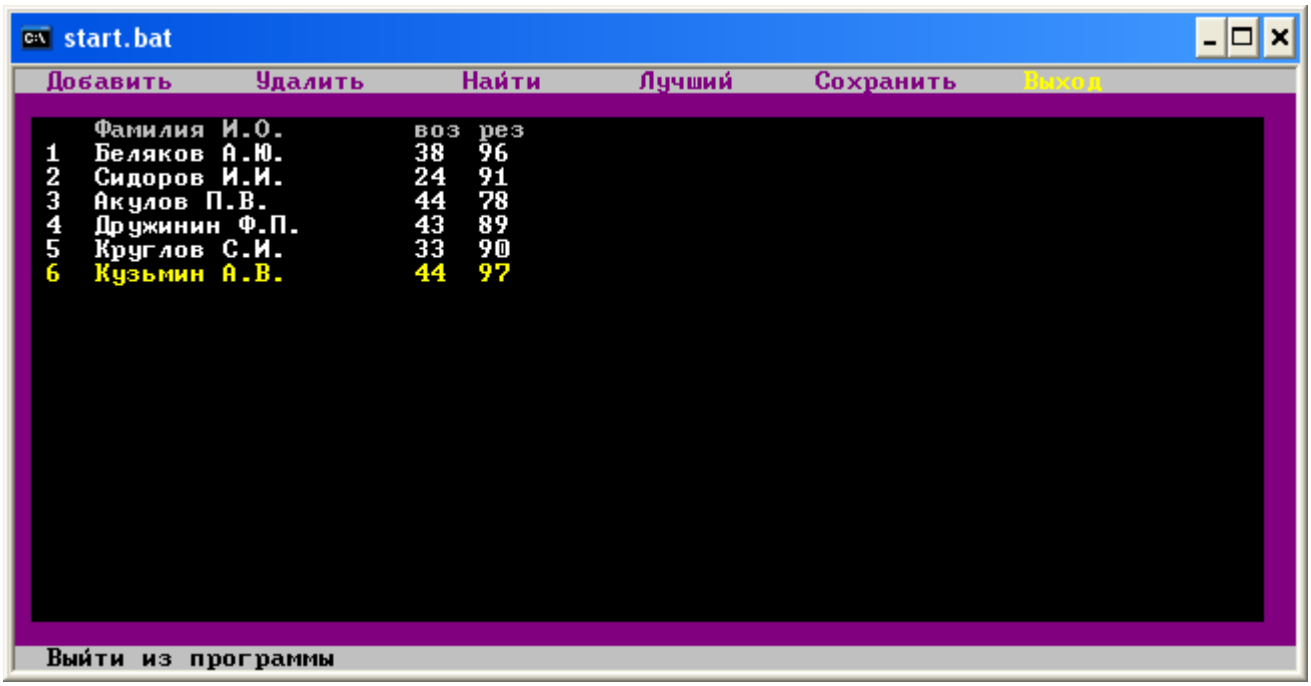


Рис.1. Главное окно программы.

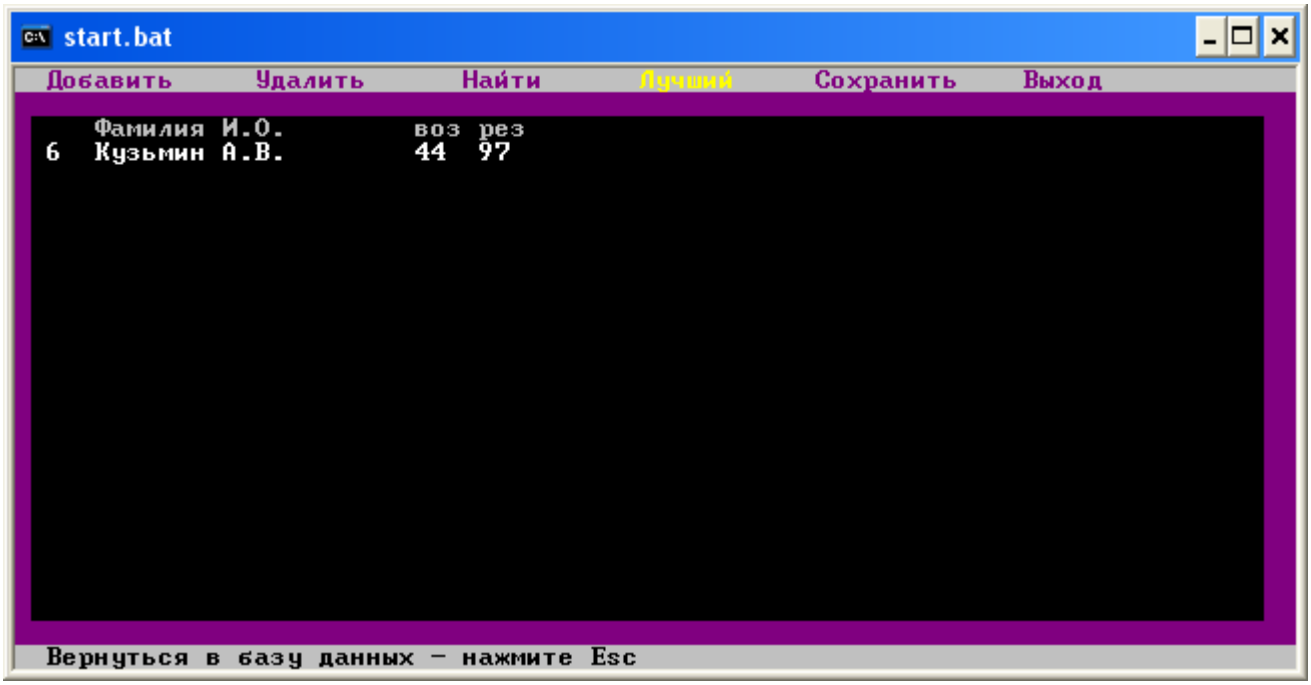


Рис.2. Демонстрация работы пункта меню «Лучший».

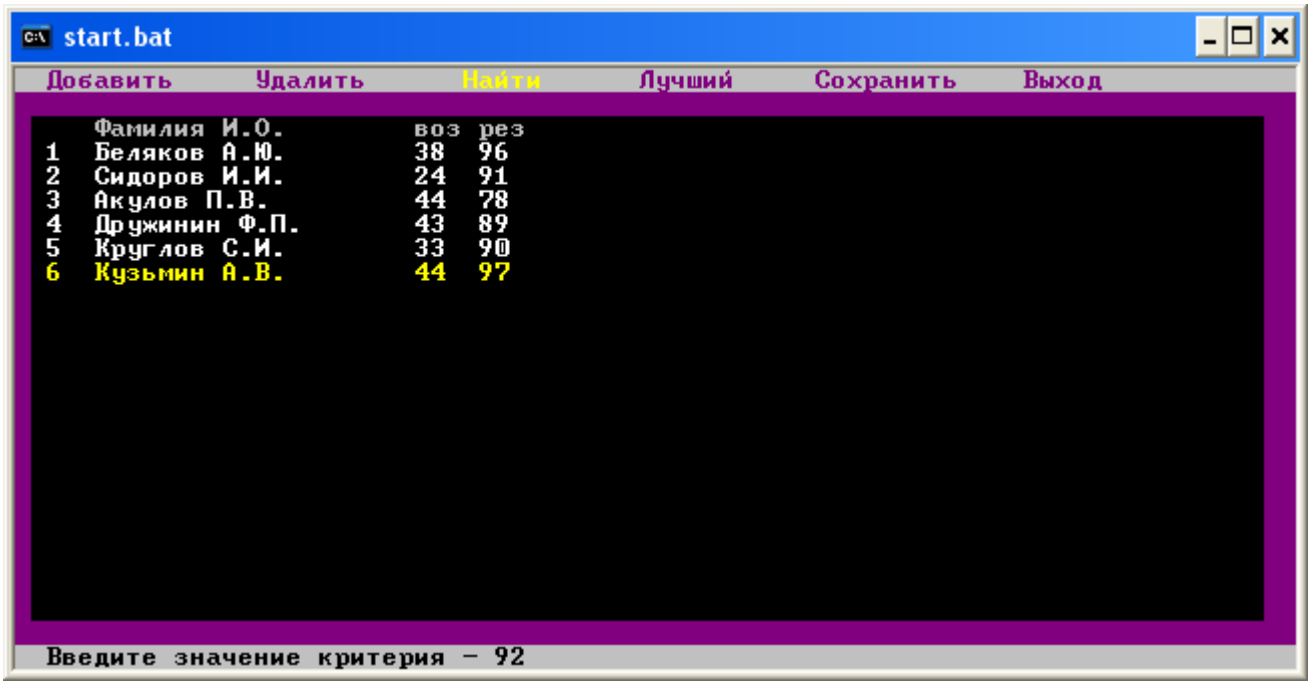


Рис.3. Демонстрация работы пункта меню «Найти».

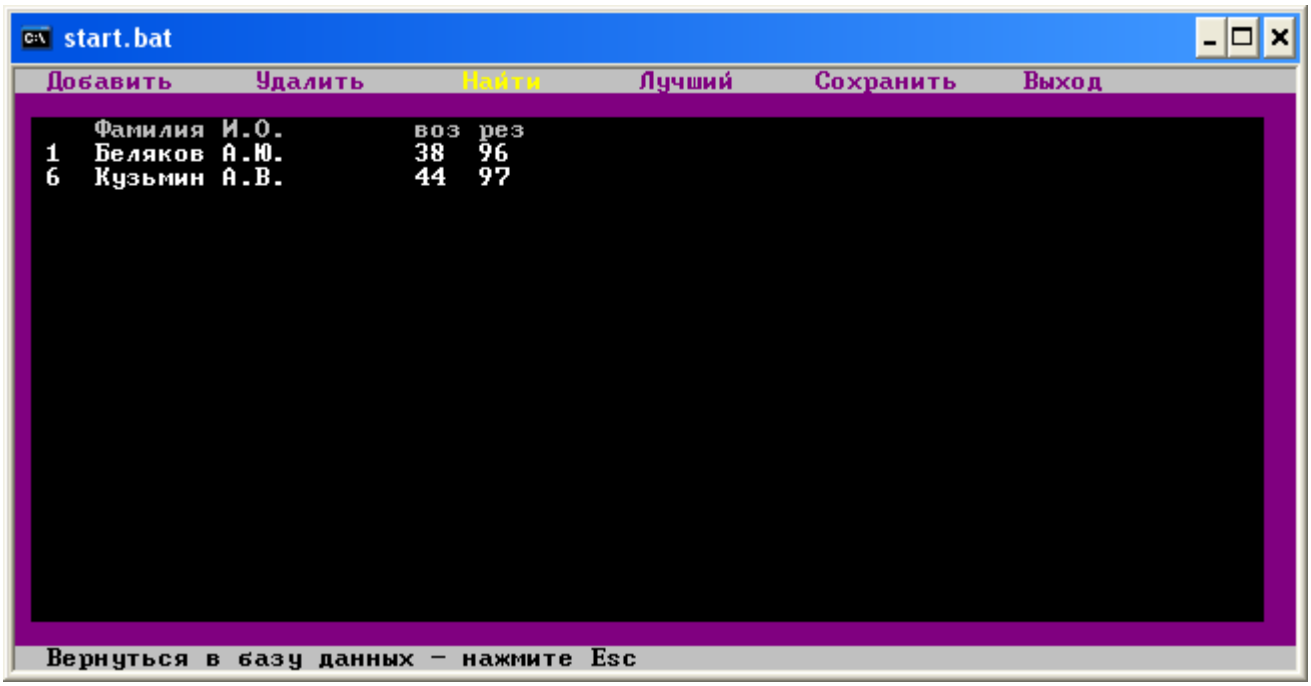


Рис.4. Результат работы пункта меню «Найти».

## Рекомендации по выполнению задания.

Что нужно уметь:

1) создавать собственные структурированные типы данных (запись);
2) обрабатывать типизированные файлы;
3) создавать многооконный интерфейс программы;
4) обрабатывать клавиатурные события;
5) работать с динамическими массивами.

По первому и второму пунктам в качестве примера рассмотрите программы из примеров 1 и 2. По третьему пункту рассмотрите программы из примеров 3 и 4.

Пример №1. Программа, в которой определяется структурированный тип данных stud (типа запись) с двумя полями, данные заносятся с клавиатуры в массив данных того же типа, затем все элементы массива сохраняются в типизированный файл.

Пример №2. Программа, в которой определяется структурированный тип данных stud (типа запись) с двумя полями, данные считываются из типизированного файла того же типа и заносятся в массив данных того же типа, затем пользователь вводит с клавиатуры критерий, а программы отбирает и выводит на экран только данные, удовлетворяющие критерию.

Пример №3. Программа, которая выводит на экран окно, в котором цвет фона и цвет шрифта выбираются случайным образом. Любое нажатие на клавиши приводит к смене цветов. Нажатие на клавишу Esc приводит к выходу из программы. Для организации текстового окна используется процедура window, аргументами которой являются координаты левого верхнего и правого нижнего углов окна. Процедура gotoxy перемещает курсор внутри последнего обозначенного окна. Процедуры textbackground и textcolor определяют цвета фона и шрифта.

Пример №4. Программа, которая выводит на экран коды нажатых клавиш, включая и управляющие (стрелки, Insert, Delete и т.д.). Обычные клавиша выдают в буфер клавиатуры один байт, а управляющие – два байта, причем код первого из них всегда равен нулю. Зная это легко узнать, нажата ли управляющая клавиша или обычная: достаточно проверить, равен ли код нулю. Но чтобы узнать, какая именно управляющая клавиша нажата, приходиться считывать код нажатой клавиши второй раз. Для обычных клавиш второго считывание не требуется. Этот пример поможет вам организовать меню. Нажимаете управляющие клавиши и двигаетесь по пунктам меню, обновляя после каждого нажатия окно процедурой window.

## Примеры программ.

Пример №1.

```
uses crt;
  type
    stud=record
    fam: string[20];
    oc: byte;
  end;
  var m: array[1..3] of stud;
    s: string;
    i: byte;
    f: file of stud;
begin  { начало программы }
  clrscr;
  assign(f,'stud.db');
  rewrite(f); { открыть файл для перезаписи }

  for i:=1 to 3 do
  begin
    write('введите фамилию студента ',i,': ');
    readln(m[i].fam);
    write('введите оценку: ');
    readln(m[i].oc);
  end;

  for i:=1 to 3 do
    write(f,m[i]); { записать в файл каждый элемент }

  close(f); { закрыть файл }
  readkey;
end.
```

Пример №2.

```
uses crt;
  type
    stud=record
    fam: string[20];
    oc: byte;
  end;
  var m: array[1..3] of stud;
    s: string;
    i,k,kol: byte;
    f: file of stud;
begin  { начало программы }
  clrscr;
  assign(f,'stud.db');
  reset(f); { открыть файл для чтения }

  kol:=1; { установить начальное значение счетчика }
  while not eof(f) do { пока не кончится файл }
  begin
    inc(kol); { увеличивай значение счетчика }
    read(f,m[kol]); { читай из файла каждый элемент }
  end;

  write('введите критерий: '); readln(k);
  writeln('везунчики');
  for i:=1 to kol do { перебрать все элементы }
    if m[i].oc>=k { выбрать только >= значения критерия }
      then writeln(m[i].fam); { вывести их на экран }

  close(f); { закрыть файл }
  readkey;
end.
```

Пример №3.

```
uses crt;

procedure okno(cvt,cvb: byte);
begin
  textbackground(cvb); { цвет фона }
  textcolor(cvt); { цвет шрифта }
  window(3,2,20,12); { создаем текстовое окно }
  clrscr; { очищаем его }
  gotoxy(2,2); { устанавливаем курсор в позицию внутри окна }
  write('окно');
end;

begin
  randomize;
  repeat
    textbackground(1); { цвет фона всего экрана синий }
    clrscr;
    okno(random(16),random(16)); { цвета выбираем случайно }
  until ord(readkey)=27; { пока не нажмем Esc }
end.
```

Пример №4.

```
uses crt;
var cod: byte;

begin
  clrscr;
  repeat
    cod:=ord(readkey); { считываем первый раз }
    case cod of
      0: cod:=ord(readkey);
            { если нажата управляющая клавиша,
              то код считываем второй раз }
      27: halt;
            { если нажата клавиша Esc,
              то выход из программы }
    end;
    writeln('код клавиши - ',cod);
  until 1=2; { бесконечный цикл }
end.
```