МИНИСТЕРСТВО ОБРАЗОВАНИЯ РФ

Кабардино-Балкарский Государственный Университет

им. Бербекова Х.М.

Выполнил: студент

Факультета Информатики и Управления

Проверил:

Нальчик 2003г.

**Содержание.**

1. Введение.

2 База данных и способы ее представление.

1. Свойства полей базы данных.
2. Цели и задачи.
3. Разработка системного меню.
4. Описание программы.

6.1 Инструкция по использованию программы.

6.2 Описание процедур.

1. Результаты.
2. Вывод.
3. Листинг программы.

**1.Введение.**

На сегодняшний день в мире работают сотни миллионов персональных

компьютеров. Ученые, экономисты, политики считают, что к началу треть-

го тысячелетия:

* количество компьютеров в мире сровняется с числом жителей развитых

стран.

* большинство этих компьютеров будет включено в мировые информацион-

ные сети.

* вся накопленная человечеством к началу третьего тысячелетия информация

будет переведена в компьютерную (двоичную) форму, а вся информация будет

готовиться при помощи (или при участии) компьютеров; вся информация будет

бессрочно храниться в компьютерных сетях;

* полноценный член общества третьего тысячелетия должен будет каждоднев

но взаимодействовать с локальными, региональными или мировыми сетями с помощью компьютеров.

При такой компьютеризации практически всех отраслей жизнедеятельно

сти человека возникает вопрос о создании программ позволяющих создавать

подобные базы данных. Поэтому и была разработана данная программа ,кото

рая позволяет создать базу данных, хранящую в себе информацию об успеваемости школьников..

#### 2.База данных и способы ее представления.

База Данных (БД) - это информация, представленная в виде двумерных таблиц. БД содержит множество строк, каждая из которых соответствует объекту. Для каждого объекта используются определенные независимые позиции, которые называются полями. Представим себе такую БД, содержащую строки и столбцы (простейший случай). Каждая строка, называемая так же записью, соответствует определенному объекту. Каждый столбец содержит значения соответствующих данных об объекте.

БД может состоять не из одной таблицы, а из двух, трех и более. Дополнительную информацию об объекте можно хранить в дополнительных таблицах.

Одно из мощных средств БД состоит в том, что информацию можно упорядочивать по тому критерию, который задает пользователь. В Pascal БД предоставляется в виде списка термов вида: имя\_предиката\_базы (поля\_записи). Имена БД описываются в разделе . Доступ к записям БД осуществляется с помощью предиката базы. pascal предоставляет довольно много средств по работе с такими БД: загрузка, запись, добавление и т.д.

*База данных – это организованная структура, предназначенная для хранения информации.* В современных базах данных хранятся не только данные, но и информация.

Это утверждение легко пояснить, если, например, рассмотреть базу данных крупного банка. В ней есть все необходимые сведения о клиентах, об их адресах, кредитной истории, состояние расчетных счетов, финансовых операциях и т.д. Доступ к этой базе данных имеется у достаточно большого количества сотрудников банка, но среди них вряд ли найдется такое лицо, которое имеет доступ ко всей базе полностью и при этом способно единолично вносить в нее произвольные изменения. Кроме данных, база содержит методы и средства, позволяющие каждому из сотрудников оперировать только с теми данными, которые входят в его компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным сотрудникам, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод и редактирование данных. С понятием базы данных тесно связано понятие *системы управления базой данных.* Это комплекс программных средств, предназначенных для создания структуры новой базы, наполнение ее содержимым, редактирование содержимого и визуализации информации. Под *визуализацией информации* базы понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройства вывода или передачи по каналам связи. В мире существует множество систем управления базами данных. Несмотря на то что они могут по-разному работать с разными объектами и предоставляют пользователю различные функции и средства, большинство СУБД опираются на единый устоявшийся комплекс основных понятий. Это дает нам возможность рассмотреть одну систему и обобщить ее понятия, приемы и методы на весь класс СУБД. В качестве такого учебного объекта мы выберем СУБД Pascal 7.0, входящую в пакет Pascal 7.0.

##### 3.Свойства полей базы данных.

Поля базы данных не просто определяют структуру базы – они еще определяют групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей. Ниже перечислены основные свойства полей таблиц баз данных на примере СУБД Pascal 7.0 ..

* Имя поля – определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц).
* Тип поля – определяет тип данных, которые могут содержаться в данном поле.
* Размер поля – определяет предельную длину (в символах) данных, которые могут размещаться в данном поле.
* Формат поля – определяет способ форматирования данных в ячейках, принадлежащих полю.
* Маска ввода – определяет форму, в которой вводятся данные а поле (средство автоматизации ввода данных).
* Подпись – определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется свойство Имя поля).
* Значение по умолчанию – то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных).
* Условие на значение – ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода, которое используется, как правило, для данных, имеющих числовой тип, денежный тип или тип даты).
* Сообщение об ошибке – текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных.
* Обязательное поле – свойство, определяющее обязательность заполнения данного поля при наполнении базы.
* Пустые строки – свойство, разрешающее ввод пустых строковых данных (от свойства Обязательное поле отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например к текстовым).
* Индексированное поле – если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. Кроме того, для индексированных полей можно сделать так, что значение в записях будут проверяться по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных.

Поскольку в разных полях могут содержаться данные разного типа, то и свойства у полей могут различаться в зависимости от типа данных. Так, например, список вышеуказанных свойств полей относится в основном к полям текстового типа.

Поля других типов могут иметь или не иметь эти свойства, но могут добавлять к ним и свои. Например, для данных, представляющих действительные числа, важным свойством является количество знаков после десятичной запятой. С другой стороны, для полей, используемых для хранения рисунков, звукозаписей, видео клипов и других объектов OLE, большинство вышеуказанных свойств не имеют смысла.

**4.Цели и задачи*.***

При создании этой программы стояли следующие цели:

* Написать программу, которая позволила бы обрабатывать, сортировать и изменять информацию о автостоянки.

Так же при создании этой программы стояли следующие задачи:

* Данная программа должна иметь простой и удобный пользовательский интерфейс.
* Данная программа должна иметь малую ресурсоёмкость.

##### 5.Разработка системного меню.

Системное меню или основное меню должно обеспечивать удобное взаимодействие пользователя с программой. В меню должны войти пункты сохранения, просмотра, ввода новых данных и.т.д. Пользователю нужно всего лишь нажать кнопку ‘enter’. В меню данной программы присутствует шесть пунктов:

1 -Создание файла

2 -Добавления запеси

3 -Корректировка запеси

4 -Просмотр запеси из файла

5 -Удаление запеси

6 -Выход

**1** - Создание нового файла – Создается новый файл с именем задаваемым пoльзователем программы

**2** - Просмотр содержимого файла – на экран поочередно выдаются раннее созданные записи в виде:

Фамилия хозяина:

Имя хозяина:

марка машины:

модель маштны:

тип кузова:

номер машины:

регион:

год выпуска:

цвет:

**3**  – Добавление записи – Создание новой записи и файле добавляя его в конец записи.

4 – Поиск по номеру палаты – Позволяет находить данные о отдыхающем по

номеру палаты , в котором зарегистрирован отдыхающий.

**5** - Выход из программы - выход из программы

**6.Описание программы.**

**6.1Инструкция по использованию программы.**

Для запуска этой программы необходимо перейти на диск 3.5“A” прилагаемый к данной работе и войти в директорию «KYRSOVAI».

Создания базы данных.

Для того чтобы создать базу данных пользователю необходимо нажать пункт «СОЗДАНИЕ ФАЙЛА» в главном меню с помощью «ENTER».После этого появится следующая строка: «ЗАДАЙТЕ ИМЯ ФАЙЛА КОТОРЫЙ БУДЕТ СОЗДОВАТЬСЯ». Далее через «ENTER» вводятся данные.

Дополнение данных.

Для дополнения данных необходимо нажать кнопку «ENTER» . После

этого появиться следующая строка: «ЗАДАЙТЕ ИМЯ ФАЙЛА В КОТОРЫЙ БУДЕТ ДОБОВЛЯТЬСЯ ЗАПИСЬ»

Выыход.

Возврощает в главную програму.

* 1. **Описание процедур.**

***Creatfile*-создание файла;**

**Addrecord-добавление запеси;**

**Inputfile-ввод данных;**

**Redzap-корректировка;**

**Openfile-просмотр;**

**Clearfile-удаление.**

**Вывод.**

Проделанная работа позволяет любому пользователю с легкостью создавать большие объемы информации, обрабатывать их, сортировать, делать выборки по определенным критериям.

Использование такой программы в современном мире значительно облегчает деятельность человека.

Program kursovay;

uses Crt,Dos;

type prokat=record

Sname\_Hozaina,Name\_Hozaina:string[15];

markauto:string[20];

model\_auto:string[20];

tip\_kyzova:string[20];

nomer\_auto:string[20];

region:integer;

god\_vipyska:integer;

color:string[20];

end;

var f:file of prokat;

zn,filename:string;

k:integer;

label 11;

{-----------------------------------------------------------}

procedure Creatfile;

begin

writeln('Задайте имя файла,который будет создаваться');

readln(filename);

assign(f,filename);

rewrite(f);

end;

{-----------------------------------------------------------}

procedure Addrecord;

var rez:integer; ch:char;

begin

while rez<>0 do

begin

writeln('Задайте имя файла,в который будет добавляться запись');

readln(filename);

assign(f,filename);

reset(f);

rez:=IOResult;

if rez<>0 then

begin

writeln('Проверьте правильность имени файла!');

writeln('Будет ли повторное задание имени файла?(Y/N)');

readln(ch);

if (ch='n') or (ch='N') then halt (0);

end;

end;

seek(f,filesize(f));

end;

{==========================================================}

procedure Inputfile;

var st:prokat;

k:char;

begin

repeat

writeln('Введите данные о новом клиенте : ');

write('Фамилия Хозяина : '); readln(st.Sname\_Hozaina);

write('Имя Хозяина:' ); readln(st.Name\_Hozaina);

write('марка машины:'); readln(st.markauto);

write('модель машины:'); readln(st.model\_auto);

write('тип кузова:'); readln(st.tip\_kyzova);

write('номер машины: '); readln(st.Nomer\_auto);

write('регион:'); readln(st.region);

write('год выпуска: '); readln(st.God\_vipyska);

write('цвет: '); readln(st.color);

write(f,st);

write('Надо ли вводить данные о новом клиенте ? (Y/N)>');readln(k);

writeln;

until(k='n') or (k='N');

close(f);

end;

{=========================================================}

procedure redzap;

var

st:prokat;

filename,str:string;

Sname\_Hozaina2,Name\_Hozaina2:string[15];

markauto2:string[20];

model\_auto2:string[20];

tip\_kyzova2:string[20];

nomer\_auto2:string[20];

region2:integer;

god\_vipyska2:integer;

color2:string[20];

x:integer;

begin

writeln('Задайте имя файла');

readln(filename);

assign(f,filename);

reset(f);

with st do

begin write('Введите фамилию клиента: ');readln(str);

x:=0;

while not eof(f) do

begin

read(f,st);

if st.Sname\_Hozaina=str then

begin

writeln('Введите новые данные о клиенте: ');

write('Фамилия хозяина: '); readln(st.Sname\_Hozaina);

write('Имя хозяина: '); readln(st.Name\_Hozaina);

write('марка машины:'); readln(st.markauto);

write('модель машины:'); readln(st.model\_auto);

write('тип кузова:'); readln(st.tip\_kyzova);

write('номер машины:'); readln(st.nomer\_auto);

write('регион:'); readln(st.region);

write('год выпуска:'); readln(st.god\_vipyska);

write('цвет: '); readln(st.color);

Sname\_Hozaina2:=Sname\_Hozaina; markauto2:=markauto;

Name\_Hozaina2:=Name\_Hozaina; color2:=color;

Nomer\_auto2:=Nomer\_auto; model\_auto2:=model\_auto;

God\_vipyska2:=God\_vipyska; tip\_kyzova2:=tip\_kyzova;

region2:=region;

seek(f,filepos(f)-1);

write(f,st); x:=1;

end; end;

close(f);

end;

if x=0 then begin writeln('Записи с такой фамилией нет!');readln end;

end;

{==============================================================}

procedure Openfile;

var rez:integer;ch:char;

begin

while rez<>0 do

begin

writeln('Задайте имя файла');

readln(filename);

assign(f,filename);

reset(f);

rez:=IOResult;

if rez<>0 then

begin

writeln('Проверьте правильность имени файла!');

writeln('Будет ли повторное задание имени файла ? (Y/N)');

readln(ch);

if (ch='n') or (ch='N') then halt (0);

end;

end;

end;

{============================================================}

procedure Outputfile;

var st:prokat;k:char;

begin

writeln('Хотите посмотреть данные о клиентах ? (Y/N)');

readln(k);

if (k='y') or (k='Y') then

while not eof(f) do

begin

read(f,st); clrscr;

writeln('Фамилия хозяина:',st.Sname\_Hozaina);

writeln('Имя хозяина:',st.Name\_Hozaina);

writeln('марка машины:',st.markauto);

writeln('модель машины:',st.model\_auto);

writeln('тип кузова:',st.tip\_kyzova);

writeln('номер машины:',st.nomer\_auto);

writeln('регион;',st.region);

writeln('год выпуска:',st.god\_vipyska);

writeln('цвет:',st.color);

writeln('Для продолжения нажмите (enter)');

readln;

end;

close(f);

end;

{========================================================}

procedure Clearfile;

var st:prokat;

studtoout:string[20];

fo:file of prokat;

begin

writeln('Задайте имя файла,из которого удаляются элементы');

readln(filename);

assign(f,filename);reset(f);

assign(fo,'newfile.dat');rewrite(fo);

writeln('Задайте фамилию клиента, запись о котором удаляется');

readln(studtoout);

while not eof(f) do

begin

read(f,st);

if st.sname\_Hozaina<>studtoout then write(fo,st);

end;

close(f);close(fo);

erase(f);

rename(fo,filename);

end;

{ОСНОВНАЯ ЧАСТЬ ПРОГРАММЫ}

Begin 11:clrscr;

writeln(' МЕНЮ:');

writeln('1:Создание файла');

writeln('2:Добавление записи в файл');

writeln('3:Корректировка записи');

writeln('4:Просмотр файла');

writeln('5:Удаление записи из файла');

writeln('6:Выход из программы');

readln(k);

case k of 1: begin Creatfile; Inputfile end;

2: begin Addrecord; Inputfile end;

3: redzap;

4: begin Openfile; Outputfile end;

5: Clearfile;

6: Halt;

else writeln('Ошибка.Повторите номер режима.')

end;

goto 11;

End.