

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

«Забайкальский государственный университет»

(ФГБОУ ВПО «ЗабГУ»)

Факультет Энергетический

Кафедра информатики, вычислительной техники и прикладной математики
(ИВТиПМ)

КУРСОВАЯ РАБОТА

по Программированию

на тему База данных продажи товаров в магазине

Выполнил студент группы ИВТ-19-1 Подгорбунский Виктор Сергеевич

Руководитель работы: доцент кафедры Соловьев Владимир Александрович

Чита

2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

«Забайкальский государственный университет»

(ФГБОУ ВПО «ЗабГУ»)

Факультет Энергетический

Кафедра информатики, вычислительной техники и прикладной математики
(ИВТиПМ)

ЗАДАНИЕ

на курсовую работу

По дисциплине Программирование

Студенту Подгорбунскому Виктору Сергеевичу

Специальности (направления подготовки) Информатика и вычислительная техника

1 Тема курсовой работы База данных продажи товаров в магазине

2 Срок подачи студентом законченной работы Июнь 2020

3 Исходные данные к работе создать программу, в которой создается база данных, содержащая сведения о продаже товаров в магазине. Сведения включают: наименование товара, объем продажи, стоимость покупки, Ф.И.О. продавца. В конце дня подводится итог. Программа должна предоставлять возможность просматривать, добавлять, удалять, копировать, хранить данные. В программе использовать объекты и модули.

Дата выдачи задания Февраль 2020г.

Руководитель курсовой работы

Задание принял к исполнению

«» 2020г.

Подпись студента /Виктор Сергеевич Подгорбунский/

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

«Забайкальский государственный университет»

(ФГБОУ ВПО «ЗабГУ»)

Факультет Энергетический

Кафедра информатики, вычислительной техники и прикладной математики
(ИВТиПМ)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

по Программированию

на тему База данных продажи товаров в магазине

Цель КР:

1. применение типов данных, изученных в течении двух семестров, простых и составных;
2. использование процедур и функций, модулей, файлов текстовых и типизированных, списков;
3. классы не используются;
4. создать минимальный оконный интерфейс на Delphi.

Календарный план

Наименование раздела курсовой работы	Месяц, неделя				
	Февраль	Март	Апрель	Май	Июнь
1. Ознакомился со структурой оформления КР	+				
2. Сбор информации по данной теме	+	+			
3. Изучение полученной информации		+	+		
4. Создание программы				+	
5. Оформление КР				+	
6. Защита КР					+

Оглавление

Введение	6
Глава 1. Типы данных и операции, реализуемые в КР.....	7
1.1 Типы данных.....	7
1.2 Операции, реализуемые в КР	7
Глава 2. Интерфейс приложения.	8
2.1 Описание компонентов.....	8
2.2 Реализация обработчиков событий.....	9
Глава 3. Тестирование созданного приложения, проверка полученных результатов	22
Заключение.....	30
Литература.....	31
Приложение	32

Введение

В сегодняшнем мире информационных технологий с каждым днем увеличивается значимость информации в самом широком смысле этого слова. Для больших объемов информации главным качеством является их структурированность, так как именно от характера структуры данных зависит скорость обработки поиска информации.

База данных – это совокупность сведений о реальных объектах, процессах, событиях или явлениях, относящихся к определённой теме или задаче, организованная таким образом, чтобы обеспечить удобное представление этой совокупности, как в целом, так и любой её части.

Все современные языки программирования содержат в себе встроенные возможности для быстрого и удобного создания Системы управления базами данных (СУБД).

СУБД- это программное обеспечение, с помощью которого можно обрабатывать, искать, передавать, хранить данные. От качества СУБД зависит эффективность работы с базой данных в целом. При работе баз данных и связанных с ними программ обработки и поиска информации необходимо учитывать специфику предметной области - одно из важнейших достоинств базы данных. Чаще всего используют специализированные Базы данных (БД) и СУБД.

В курсовой работе будет создана простейшая база данных продажи товаров в магазине, состоящая из одной таблицы. Для создания приложения будут использованы компоненты MainMenu, OpenFileDialog, StringGrid, сортировка и поиск по наименованию товара.

Глава 1. Типы данных и операции, реализуемые в КР.

1.1 Типы данных.

В данной курсовой работе используются следующие типы данных:

Record-это структура, состоящая из фиксированного числа компонент, называемых полями. Здесь magaz-имя типа, а x-имя переменной. Переменная x состоит из 6 полей: NazTov, ObTov, StoimPok, Fam, Nam, Otch, имеющих тип данных NazTov-string; ObTov-integer; StoimPok-real; Fam, Nam, Otch-string.

Переменная PUzel имеет тип - ^Zp-тип указателя на узел Zp. Zp – тип узла.

Переменная FZap имеет тип file of magaz- файловый тип для хранения данных

```
type
magaz = record                               //Тип магазин
    NazTov:string[20];                       //Наименование товара
    ObTov:Integer;                           //Объем продажи
    StoimPok:Real;                           //Стоимость покупки
    Fam,Nam,Otch:string[20];                //ФИО продавца
end;
PUzel = ^Zp;                                //Тип указателя на узел
Zp = record                                  //Тип узла списка
    x: magaz;                                //Сведения о магазине
    next: PUzel;                             //Указатель на следующий узел
    pred: PUzel;                             //Указатель на предыдущий узел
end;
FZap = file of magaz; //файловый тип для хранения данных о магазине
```

1.2 Операции, реализуемые в КР

```
procedure AddFirst(var f: PUzel; a: PUzel); {Вставить узел а первым в список}
procedure AddAfter(var old:PUzel; a: PUzel); {Вставить узел а после old}
procedure BuildSpisok(var f: PUzel);{Построить список; f-указатель на голову списка}
procedure WriteSpText(var f: PUzel;var ftxt:Text);{Вывод списка в текстовый файл}
procedure DelFirstElement(var f,a: PUzel); {Выделить из списка первый узел
и вернуть его пользователю}
procedure DelSpisok(var f: PUzel); //Удалить список
procedure WriteSpTip(var f: PUzel; var ftip: FZap); //запись в типизированный файл
procedure BuildTip(var ftip: FZap);{Вывод списка в типизированный файл}
procedure Result(var t: PUzel; var ftxt: Text); {Выводит итоговый результат}
```


2.2 Реализация обработчиков событий

//Процедура открытия текстового файла

```
procedure TForm1.OpenTxtFile1Click(Sender: TObject);
```

```
var
```

```
  s: string;
```

```
begin
```

```
  if not OpenFileDialog1.Execute then exit;
```

```
  s := OpenFileDialog1.FileName;
```

```
  AssignFile(ftxt,s);
```

```
  Append(ftxt);
```

```
end;
```

//Процедура закрытия текстового файла

```
procedure TForm1.CloseTxtFile1Click(Sender: TObject);
```

```
begin
```

```
  CloseFile(ftxt);
```

```
  ShowMessage('Вы закрыли текстовый файл');
```

```
end;
```

//Процедура открытия типизированного файла

```
procedure TForm1.OpenTipFile1Click(Sender: TObject);
```

```
var
```

```
  s: string;
```

```
begin
```

```
  if not OpenFileDialog1.Execute then exit;
```

```
  s := OpenFileDialog1.FileName;
```

```
  AssignFile(ftip,s);
```

```
  reset(ftip);
```

```
  seek(ftip,filesize(ftip));
```

end;

//Процедура закрытия типизированного файла

procedure TForm1.CloseTipFile1Click(Sender: TObject);

begin

 CloseFile(ftip);

 ShowMessage('Вы закрыли типизированный файл');

end;

//Процедура для создания названий столбцов

procedure TForm1.FormCreate1Click(Sender: TObject);

begin

 StringGrid1.Cells[0,0]:='№';

 StringGrid1.Cells[1,0]:='Наименование товара';

 StringGrid1.Cells[2,0]:='Объем продажи';

 StringGrid1.Cells[3,0]:='Стоимость покупки';

 StringGrid1.Cells[4,0]:='ФИО продавца';

end;

//Процедура заполнения списка

procedure TForm1.BuildList1Click(Sender: TObject);

begin

 BuildSpisok(PList);

end;

//Процедура сохранения списка в текстовый файл

procedure TForm1.SaveListTxt1Click(Sender: TObject);

 var

 s: string;

begin

```

        writeln(ftxt,'Содержимое базы данных: ');
        writeln(ftxt);
        WriteSpText(PList,ftxt);
        writeln(ftxt,' ');
        ShowMessage('Список сохранён в текстовый файл');
    end;

//Процедура заполнения списка данными из типизированного файла
procedure TForm1.BuildtipList1Click(Sender: TObject);
var
    s:string;
begin
    BuildTip(ftip);
end;

//Процедура сохранения списка в типизированный файл
procedure TForm1.SaveTipList1Click(Sender: TObject);
var
    s:string;
begin
    WriteSpTip(PList,ftip);
    ShowMessage('Список сохранён в типизированный файл');
end;

//Процедура выхода из программы
procedure TForm1.Exit1Click(Sender: TObject);
begin
    Close;
end;

```

//Процедура удаления списка

procedure TForm1.DelList1Click(Sender: TObject);

var

i,j:integer;

begin

for i:=1 to StringGrid1.RowCount-1 do

for j:=0 to StringGrid1.ColCount-1 do

begin

StringGrid1.Cells[j, i]:='';

end;

DelSpisok(PList);

end;

//Процедура вывода информации за день

procedure TForm1.Result1Click(Sender: TObject);

begin

Result(PList, ftxt);

end;

procedure TForm1.SearchClick(Sender: TObject);

var s:string;

begin

if InputQuery('Поиск', 'Введите название продукта для поиска', s) then

begin

if s<>'' then

begin

searchNameTovar(PList,s);

end

else

begin

showmessage('Данные не были введены');

```
        end;  
    end  
else  
    begin  
        exit;  
    end;  
end;  
end;
```

```
procedure TForm1.SortClick(Sender: TObject);  
var  
    SlSort, SlRow : TStringList;  
    i, j, aCol : Integer;  
begin  
    aCol := StrinGgrid1.FixedCols;  
  
    SlSort := TStringList.Create;  
    for i := StrinGgrid1.FixedRows to StrinGgrid1.RowCount - 1 do begin  
        SlRow := TStringList.Create;  
        SlRow.Assign(StrinGgrid1.Rows[i]);  
        SlSort.AddObject(StrinGgrid1.Cells[aCol, i], SlRow);  
    end;  
    SlSort.Sort;  
    j := 0;  
    for i := StrinGgrid1.FixedRows to StrinGgrid1.RowCount - 1 do begin  
        SlRow := Pointer(SlSort.Objects[j]);  
        StrinGgrid1.Rows[i].Assign(SlRow);  
    end;  
end;
```

```

    SIRow.Free;
    Inc(j);
end;
SI Sort.Free;
end;

procedure AddFirst(var f: PUzel; a: PUzel);
begin
    a^.next := f;
    if f <> nil then f^.pred := a;
    f:=a;
end;

procedure AddAfter(var old:PUzel; a: PUzel);
begin
    a^.next := old^.next;
    old^.next := a;
    if a^.next <> nil then a^.next^.pred := a;
    a^.pred := old;
end;

procedure BuildSpisok(var f: PUzel);
var
    a,d :PUzel;
    ch: char;
    i: integer;
begin
    i:=1;
while Form1.StringGRid1.Cells[1,i]<>" do
    begin

```

```

    inc(i);
end;
f:= nil;
repeat
    new(a);
    with a^.x do
        begin
            Form1.StringGrid1.Cells[0, i] := IntToStr(i);
            NazTov := InputBox('Наименование товара', '', '');
            Form1.StringGrid1.Cells[1, i] := a^.x.NazTov;
            ObTov := StrToInt(InputBox('Объем продажи', '', ''));
            Form1.StringGrid1.Cells[2, i] := IntToStr(a^.x.ObTov);
            StoimPok:= StrToFloat(InputBox('Стоимость покупки', '', ''));
            Form1.StringGrid1.Cells[3, i] := FloatToStr(a^.x.StoimPok);
            Fam := InputBox('Введите фамилию продавца', '', '');
            Nam := InputBox('Введите имя продавца', '', '');
            Otch := InputBox('Введите отчество продавца', '', '');
            Form1.StringGrid1.Cells[4, i] := a^.x.Fam + ' ' + a^.x.Nam + ' ' + a^.x.Otch;
        end;
        inc(i);
    end;
    a^.next := nil;
    a^.pred := nil;
    if (f = nil) then
        begin
            AddFirst(f,a);
            d:= f;
        end
    else
        begin
            AddAfter(d,a);

```

```

    d := a;
  end;
  ch:= InputBox('Для завершения ввода нажмите Y','')[1];
  until (ch = 'Y') or (ch ='y');
end;

```

```

procedure WriteSpText(var f: PUzel; var ftxt:Text);
var
  p: PUzel;
  s: string;
  y: magaz;
begin
  p:= f;
  while not(p = nil) do
    begin
      y:= p^.x;
      s:=y.NazTov + ' ' + IntToStr(y.ObTov) + ' ' + FloatToStr(y.StoimPok) + ' '+y.Fam
+ ' ' + y.Nam + ' ' + y.Otch;
      writeln(ftxt, s);
      p:= p^.next;
    end;
  end;
end;

```

```

procedure DelFirstElement(var f,a: PUzel);
begin
  a := f;
  f := f^.next;
  a^.next := nil;
  if f<>nil then f^.pred := nil;

```


end;

procedure DelElement(var old,a: PUzel);

begin

if (old^.next = nil) then a:= nil //old последний узел в списке

else

if (old^.next^.next = nil) then //old предпоследний узел в списке

begin

a := old^.next;

a^.pred:= nil;

old^.next:= nil;

end

else

begin //за old не менее двух узлов в списке

a := old^.next;

old^.next := a^.next;

old^.next^.pred:= old;

a^.next := nil;

a^.pred:= nil;

end;

end;

procedure DelSpisok(var f: PUzel); //Удалить список

var

a: PUzel;

begin

while (f <> nil) do

begin

DelFirstElement(f,a);

Dispose(a);

```

    end;
end;

procedure WriteSpTip(var f: PUzel; var ftip: FZap);
var
    p: PUzel;
begin
    p := f;
    while not (p = nil) do
        begin
            write(ftip, p^.x);
            p := p^.next;
        end;
    end;
end;

procedure BuildTip(var ftip: FZap);
var
    a: PUzel;
    i: integer;
begin
    seek(ftip, 0);
    i := 1;
    while Form1.StringGRid1.Cells[1, i] <> " do
        begin
            inc(i);
        end;
    while not eof(ftip) do
        begin
            New(a);
            read(ftip, a^.x);

```

```

if PList = nil then
begin
  AddFirst(PList, a);
  Form1.StringGrid1.Cells[0, i] := IntToStr(i);
  Form1.StringGrid1.Cells[1, i] := a^.x.NazTov;
  Form1.StringGrid1.Cells[2, i] := IntToStr(a^.x.ObTov);
  Form1.StringGrid1.Cells[3, i] := FloatToStr(a^.x.StoimPok);
  Form1.StringGrid1.Cells[4, i] := a^.x.Fam + ' ' + a^.x.Nam + ' ' + a^.x.Otch;
  inc(i);
end
else
begin
  AddAfter(PList, a);
  Form1.StringGrid1.Cells[0, i] := IntToStr(i);
  Form1.StringGrid1.Cells[1, i] := a^.x.NazTov;
  Form1.StringGrid1.Cells[2, i] := IntToStr(a^.x.ObTov);
  Form1.StringGrid1.Cells[3, i] := FloatToStr(a^.x.StoimPok);
  Form1.StringGrid1.Cells[4, i] := a^.x.Fam + ' ' + a^.x.Nam + ' ' + a^.x.Otch;
  inc(i);
end;
end;
end;
procedure Result(var t: PUzel; var ftxt: Text);
var
  p: PUzel;
  s:real;
begin
  s:=0;
  writeln(ftxt,"");
  p:=PList;

```

```

begin
  if p^.next=nil then
    begin
      s:=s+(p^.x.ObTov)*p^.x.StoimPok;
    end;
  while not(p=nil) do
    begin
      begin
        s:=s+(p^.x.ObTov)*p^.x.StoimPok;
      end;
      p:=p^.next;
    end;
  writeln(ftxt,'Сумма продажи за день: ',FloatToStr(s),' ');
  writeln(ftxt);
  if p=nil then
    begin
      p:=PList;
    end;
  end;
  ShowMessage(Итоговая сумма продажи за день: '+FloatToStr(s)+' ');
end;

```

```

procedure searchNameTovar (var a:PUzel; var s:string);
var p:PUzel;
    st,t:string;
    i:integer;
begin
  p:=a;
  while p<>nil do
    begin

```


```

if p^.x.NazTov = s then
  begin
    st:=st+('Название продукта: '+p^.x.NazTov +' | '+'Объем продажи: '+
IntToStr(p^.x.ObTov)+
    ' | '+'Стоимость покупки: '+ FloatToStr(p^.x. StoimPok)+' | '+'ФИО
продавца: '+p^.x.Fam+' '+'p^.x.Nam+' '+'p^.x.Otch+';'+'#13 ');
    end;
    p:=p^.next;
  end;
  if st="" then
    begin
      ShowMessage('Совпадений не найдено');
    end
  else
    ShowMessage('Результаты поиска по запросу "Название продукта":
'+s+'#13+st);
  end;

```


Глава 3. Тестирование созданного приложения, проверка полученных результатов

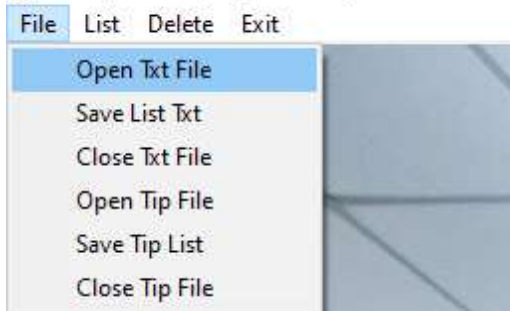
Запускаем проект на выполнение: команда Run|Run, или F9.

 База данных продажи товаров в магазине

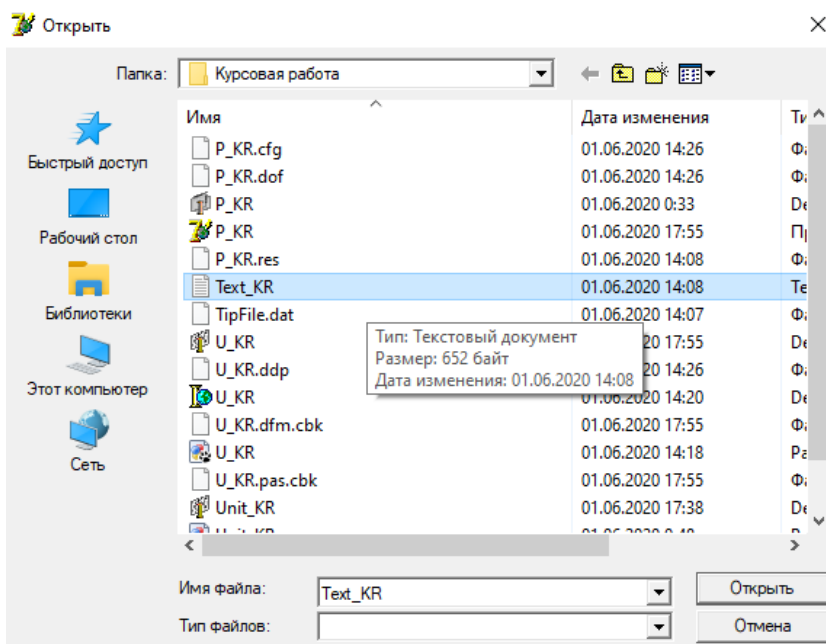
File List Delete Exit

Теперь последовательно выполним следующие операции:

 База данных продажи товаров в магазине




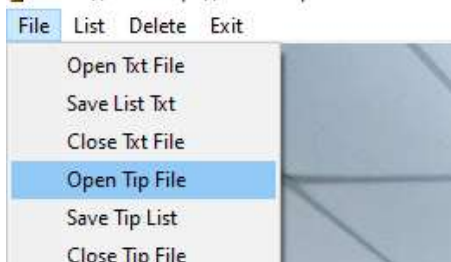
Получим



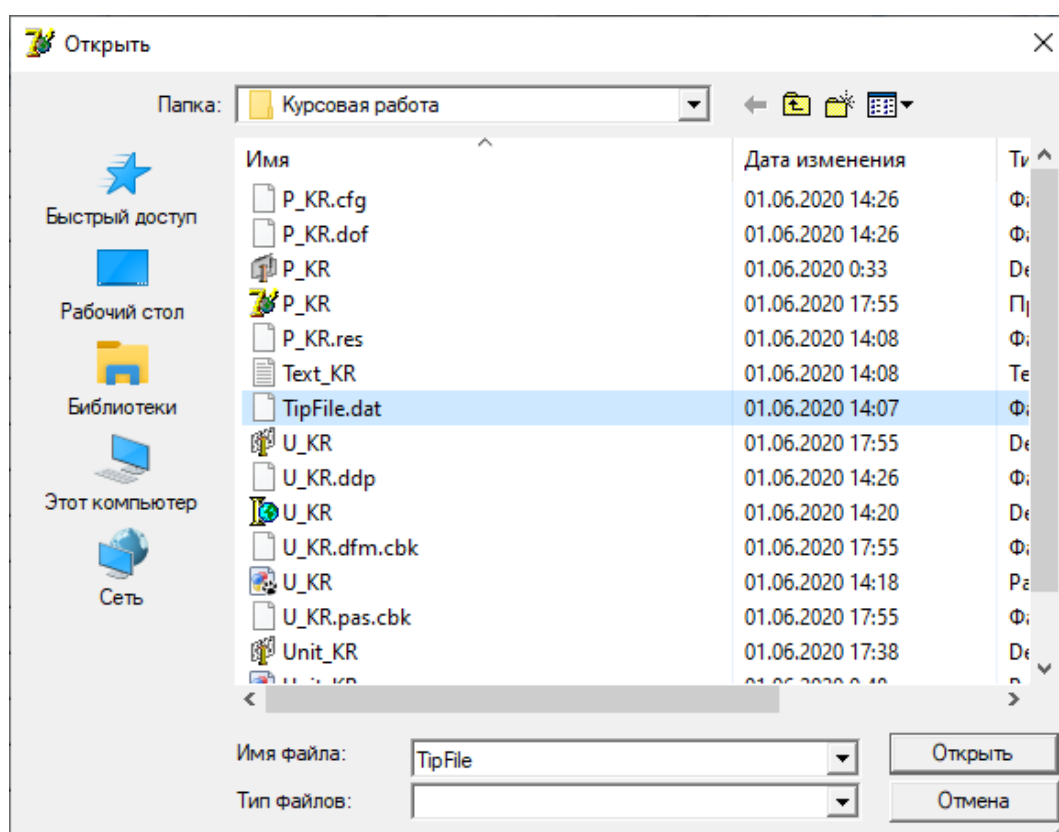
т.е найдём нужную папку, в ней текстовый файл и откроем его.

Следующая операция – открытие типизированного файла

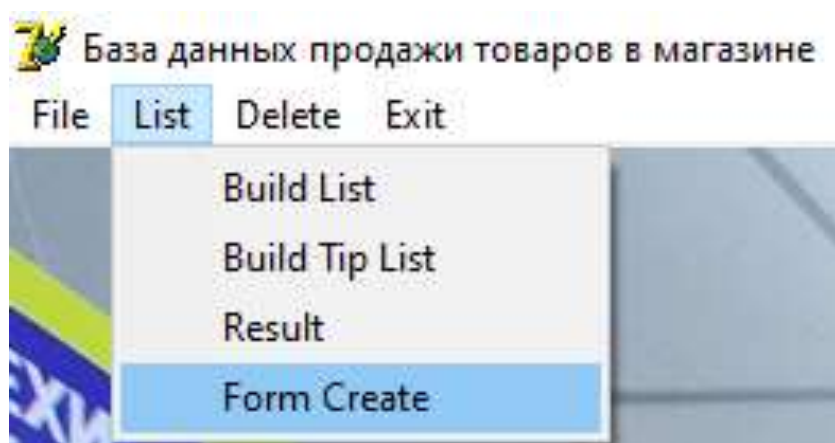
 База данных продажи товаров в магазине



Получим



Следующая операция – заполнения названия столбцов: List|Form Create

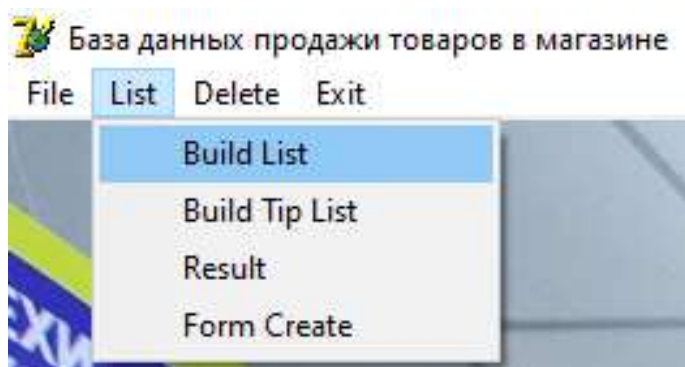


Получим

Nº	Наименование товара	Объем продажи	Стоимость покупки	ФИО продавца
----	---------------------	---------------	-------------------	--------------

Следующая операция – создание списка: можно создать вручную или заполнить список из типизированного файла:

Вручную – List|Build List



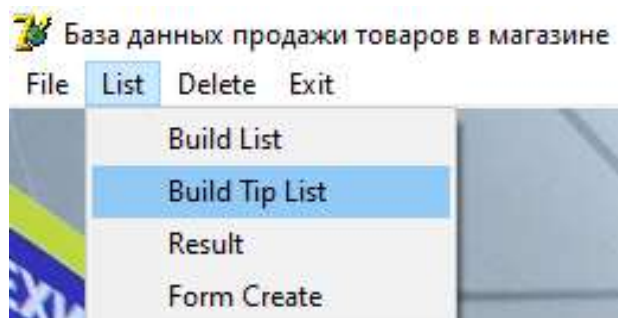
После выполнения команды циклически будут появляться окна для заполнения списка:

The image displays a sequence of five input dialog boxes, each with a title bar, a text input field, and "OK" and "Cancel" buttons. The titles of the dialog boxes are: "Наименование товара" (Goods name), "Объем продажи" (Sales volume), "Стоимость покупки" (Purchase cost), "Введите фамилию продавца" (Enter seller's surname), "Введите имя продавца" (Enter seller's name), and "Введите отчество продавца" (Enter seller's patronymic). The last dialog box has a title bar that reads "Для завершения ввода нажмите Y" (Press Y to finish input).

В первом окне вводим наименование товара, буквами; Во втором окне вводим объем продажи, целое число; В третьем окне вводим стоимость покупки, реальное число, целая и дробная части разделяются запятой; В четвертом, пятом и шестом

окне вводим ФИО продавца, буквами; В последнем окне для продолжения ввода можно нажать любую клавишу, кроме Y.

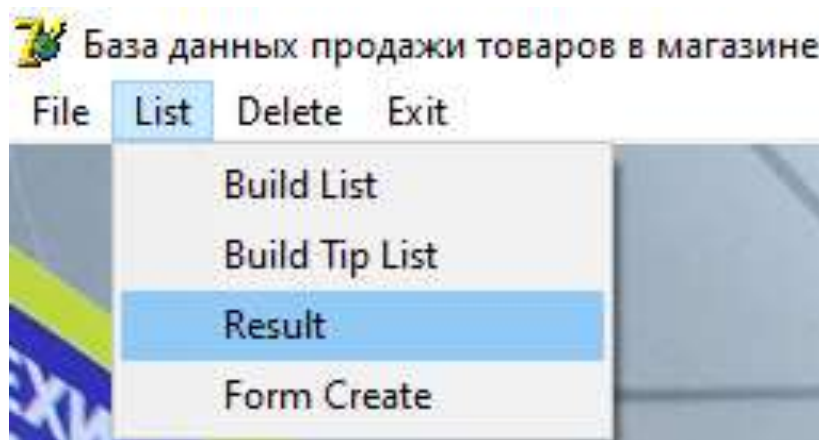
Заполнить список из типизированного файла – List|Build Tip List



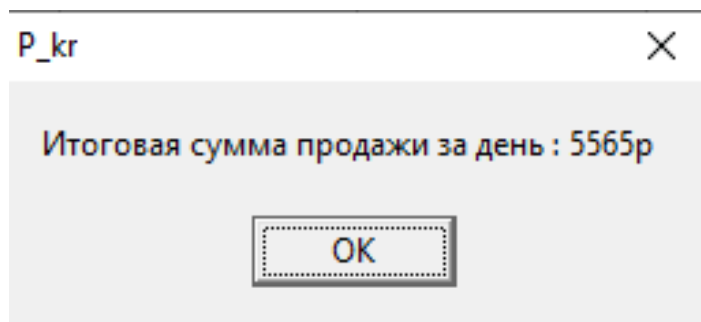
После выполнения команды получим

№	Наименование товара	Объем продажи	Стоимость покупки	ФИО продавца
1	Хлеб	20	30	Петров Пётр Петрович
2	Молоко	10	68	Петров Пётр Петрович
3	Масло	3	75	Петров Пётр Петрович
4	Яйца	9	100	Сидоров Сидор Сидорович
5	Макароны	5	92	Петров Пётр Петрович
6	Яблоки	18	150	Сидоров Сидор Сидорович

Чтобы вывести информацию, нажимаем – List|Result



Получаем



Чтобы найти наименование товара – SearchTovar

База данных продажи товаров в магазине

File List Delete SearchTovar Sortirovka Exit

Вводим наименование товара например «Хлеб»

Поиск

Введите наименование продукта для поиска

OK Cancel

Получаем

P_kr

Результаты поиска по запросу "Наименование товара": Хлеб
Наименование товара: Хлеб | Объем продажи: 20 | Стоимость покупки: 30 | ФИО продавца: Петров Пётр Петрович

OK

Чтобы отсортировать список по алфавиту – Sort

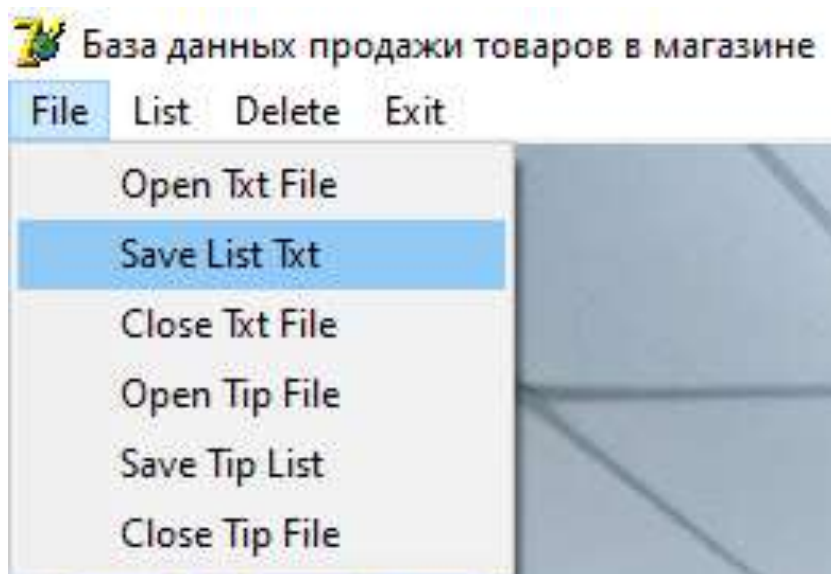
База данных продажи товаров в магазине

File List Delete SearchTovar Sort Exit

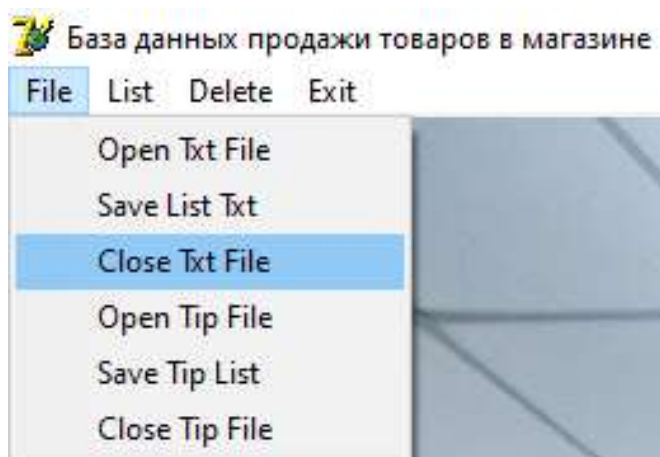
Получаем

№	Наименование товара	Объем продажи	Стоимость покупки	ФИО продавца
5	Макароны	5	92	Петров Пётр Петрович
3	Масло	3	75	Петров Пётр Петрович
2	Молоко	10	68	Петров Пётр Петрович
1	Хлеб	20	30	Петров Пётр Петрович
6	Яблоки	18	150	Сидоров Сидор Сидорович
4	Яйца	9	100	Сидоров Сидор Сидорович

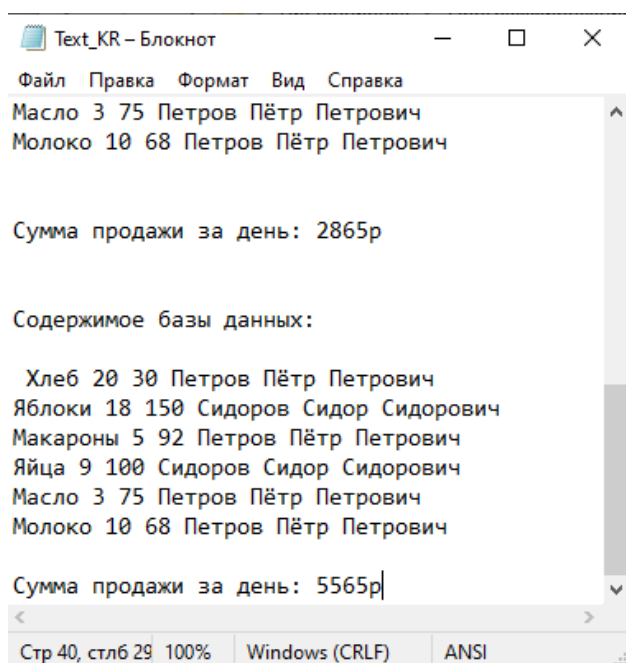
Чтобы сохранить список в текстовый файл – File|Save List Txt



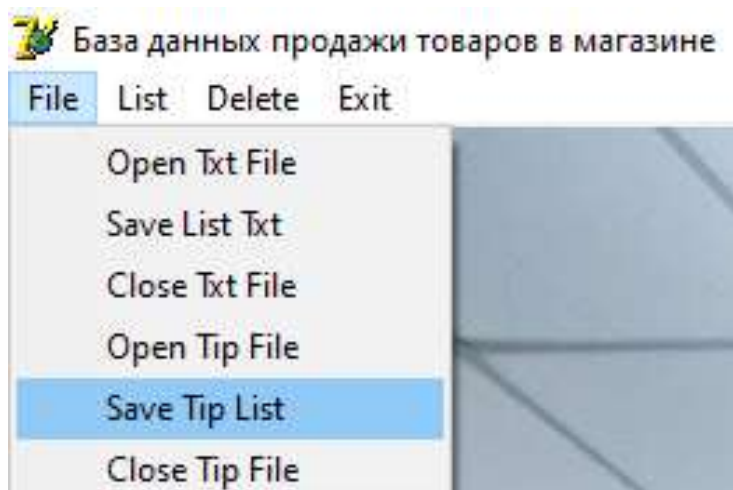
И закрываем



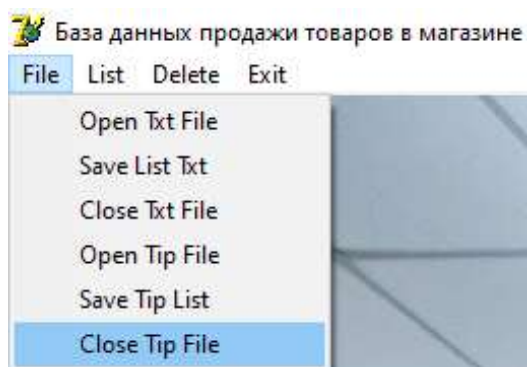
В итоге



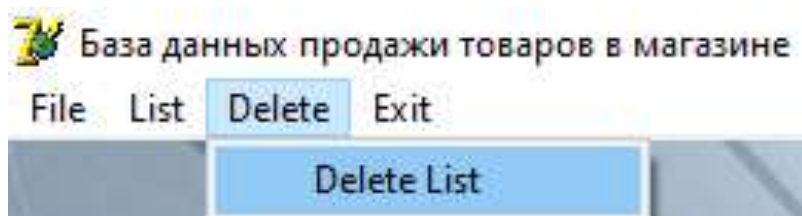
Чтобы сохранить список в типизированный файл – File|Save Tip List



И так же закрываем – File|Close Tip File




Чтобы удалить список – Delete|



Получаем

[illegible]

Чтобы закрыть приложение – Exit

 База данных продажи товаров в магазине

File List Delete Exit

Заключение

Курсовая работа была выполнена в соответствии поставленному заданию и создана в среде DELPHI 7. В ходе ее выполнения была разработана программа, в которой создается база данных, содержащая сведения о продаже товаров в магазине. В процессе выполнения курсовой работы были получены навыки работы с программой Delphi, изучена и проанализирована соответствующая литература, использованы четыре источника информации, самостоятельно изучены компоненты Delphi: MainMenu, OpenDialog, StringGrid.

В курсовой работе самостоятельно изученны и успешно были применены сортировка и поиск по наименованию товара. Данная работа может быть использована студентами для обучения.

Литература

1. А.Я. Архангельский. Язык Pascal и основа программирования в Delphi. Учебное пособие – М.: ООО«Бином-Пресс», 2004г.-496с.
2. Онлайн учебник по Delphi 7 <https://delphi.support.uz/>
3. Понамарев, В. Базы данных в Delphi 7. Самоучитель; СПб: Питер - М., **2015**. - 224 с.
4. Источник из сети интернет. Форма, Компоненты и их свойства.
<http://dppc.ru/data/attachments/library/lektsyja-3-4komponent-27253.pdf>

Приложение

Код интерфейсной части модуля формы:

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Menus, ExtCtrls, StdCtrls, Grids, Unit_KR, jpeg;

type

TForm1 = class(TForm)

MainMenu1: TMainMenu;

OpenDialog1: TOpenDialog;

File1: TMenuItem;

List1: TMenuItem;

FormCreate1: TMenuItem;

BuildList1: TMenuItem;

SaveListTxt1: TMenuItem;

BuildtipList1: TMenuItem;

SaveTipList1: TMenuItem;

Result1: TMenuItem;

Delete1: TMenuItem;

DelList1: TMenuItem;

Exit1: TMenuItem;

StringGrid1: TStringGrid;

OpenTxtFile1: TMenuItem;

OpenTipFile1: TMenuItem;

CloseTxtFile1: TMenuItem;

CloseTipFile1: TMenuItem;

Image1: TImage;

Search: TMenuItem;

Sort: TMenuItem;


```

    procedure OpenTxtFile1Click(Sender: TObject); //Процедура открытия
текстового файла

    procedure CloseTxtFile1Click(Sender: TObject); //Процедура закрытия
текстового файла

    procedure OpenTipFile1Click(Sender: TObject); //Процедура открытия
типизированного файла

    procedure CloseTipFile1Click(Sender: TObject); //Процедура закрытия
типизированного файла

    procedure FormCreate1Click(Sender: TObject); //Процедура для создания
названий столбцов

    procedure BuildList1Click(Sender: TObject); //Процедура заполнения списка

    procedure SaveListTxt1Click(Sender: TObject); //Процедура сохранения списка
в текстовый файл

    procedure BuildtipList1Click(Sender: TObject); //Процедура заполнения списка
данными из типизированного файла

    procedure SaveTipList1Click(Sender: TObject); //Процедура сохранения списка
в типизированный файл

    procedure Exit1Click(Sender: TObject); //Процедура выхода из программы

    procedure DelList1Click(Sender: TObject); //Процедура удаления списка

    procedure Result1Click(Sender: TObject); //Процедура вывода информации
за день

    procedure SearchClick(Sender: TObject);

    procedure SortClick(Sender: TObject);

private
    { Private declarations }

public
    { Public declarations }

end;

```

```

var
  Form1: TForm1;
  PList: PUzel;      //указатель на список
  ftxt: text;        //файловая переменная
  ftip : FZap; // FZap – тип типизированного файла, определён выше

```

Код интерфейсной части созданного модуля:

```

uses SysUtils, Dialogs, Controls;      //Модули для ввода/вывода данных

```

```

type

```

```

magaz = record                        //Тип магазин
  NazTov:string[20];                //Наименование товара
  ObTov:Integer;                    //Объем продажи
  StoimPok:Real;                    //Стоимость покупки
  Fam,Nam,Otch:string[20];         //ФИО продавца
end;

```

```

PUzel = ^Zp;                        //Тип указателя на узел

```

```

Zp = record                          //Тип узла списка
  x: magaz;                          //Сведения о магазине
  next: PUzel;                       //Указатель на следующий узел
  pred: PUzel;                       //Указатель на предыдущий узел
end;

```

```

FZap = file of magaz; //Файловый тип для хранения данных о магазине

```

```

procedure AddFirst(var f: PUzel; a: PUzel); {Вставить узел а первым в список}

```

```

procedure AddAfter(var old:PUzel; a: PUzel); {Вставить узел а после old}

```

```

procedure BuildSpisok(var f: PUzel);{Построить список; f-указатель на голову
списка}

```

```
procedure WriteSpText(var f: PUzel; var ftxt: Text); {Вывод списка в текстовый  
файл}
```

```
procedure DelFirstElement(var f, a: PUzel); {Выделить из списка первый узел  
и вернуть его пользователю}
```

```
procedure DelSpisok(var f: PUzel); //Удалить список
```

```
procedure WriteSpTip(var f: PUzel; var ftip: FZap); //запись в типизированный  
файл
```

```
procedure BuildTip(var ftip: FZap); {Вывод списка в типизированный файл}
```

```
procedure Result(var t: PUzel; var ftxt: Text); {Выводит итоговый результат}
```

```
procedure searchNameTovar (var a: PUzel; var s: string); {поиск по названию  
товара}
```