



## Компонент – Тmemo.

Вкладка **standart**.

**Name** – (memo).

**Text** – все содержимое компонента независимо от количества заполненных строк.

**Lines** – отдельные строки компонента. Свойство можно задавать через инспектор объектов или через программу. Все строки нумеруются, начиная с нуля.

**Memo1.lines[n]:=<выражение>;** //присваивает значение n-му элементу (элемент существует!).

**Memo1.lines.clear;** //очищает весь компонент.

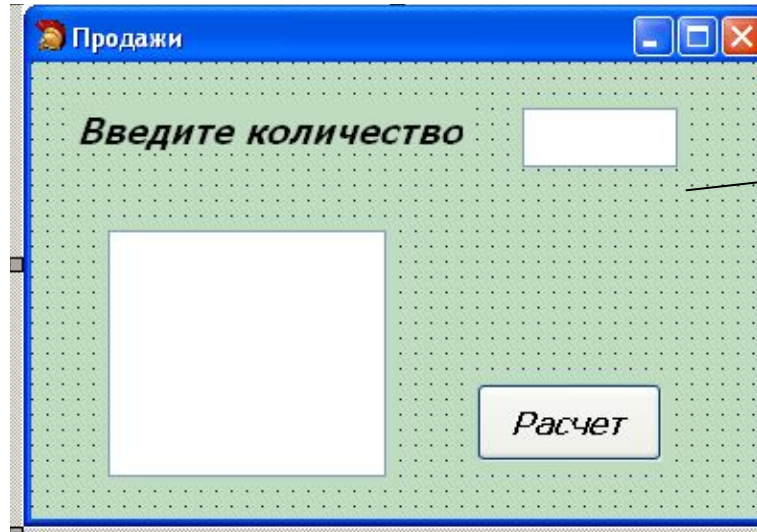
**Memo1.lines.add('строка');** //добавляет новую строку в компонент.

**Memo1.lines.loadfromfile('имя');** // загружает в компонент данные из текстового файла.

**Memo1.lines.savetofile('имя');** //сохраняет данные из компонента в текстовом файле. (имя – полное имя файла)

<b>Turbo Pascal</b>	<b>Delphi</b>
Var F: text;	Var F: textfile;
Assign (f, 'полное имя файла');	Assignfile (f, 'полное имя файла');
Close(f);	Closefile(f);

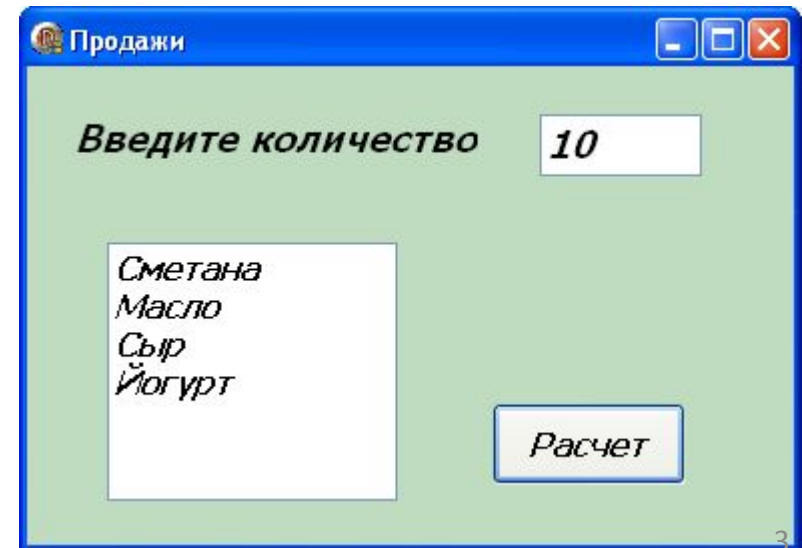
Задача: дан текстовый файл, содержащий информацию о продаже товаров: товар, количество. Вывести в поле мемо все товары, количество которых больше введенного значения в поле редактирования.



Label1

edit1

button1



Молоко  
5  
Сметана  
11  
Масло  
23  
Сыр  
34  
Творог  
8  
Йогурт  
23  
Кефир  
9

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    Memo1: TMemo;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
  20
  товар=record
    nt:string[15];
    kol: integer;
  end;

var
  Form1: TForm1;
  tov: товар;
  f: textfile;

implementation
```

```
·  implementation
30  ·  {$R *.dfm}
·  // Заполнение списка
·  procedure TForm1.Button1Click(Sender: TObject);
·  begin
-   assignfile(f, 'G:\студенты\2 курс\memo\tovar.txt');
·   reset(f);
·   memo1.Lines.Clear;
·   if edit1.text='' then begin
·   showmessage('ВВЕДИТЕ КОЛИЧЕСТВО'); exit;end;
40  while not eof(f) do begin
·   readln(f,tov.nt);
·   readln(f,tov.kol);
·   if tov.kol>strtoint(edit1.Text) then memo1.Lines.Add(tov.nt);
·   end;
-   closefile(f);
·   end;
·
·
·  end.
```

49

# Работа со списками

## Вкладка **standart**



**TListBox**



**TComboBox**

## Основные свойства

**name** – имя, используемое в программе `listbox1` или `combobox1` соответственно.

**items** (элементы списка) – может устанавливаться как в программе, так и инспекторе объектов. Определяет значения элементов списка.

**Itemindex** - номер выбранного элемента списка. Номер первого элемента списка равен 0, если не выбран ни один элемент номер равен -1.

Например,

**Case `listbox1.itemindex` of**

**0: команда;**

**1: команда;**

**2: и т.д.;**

**End;**

**listbox1.items[n]:=<выражение>;** //присваивает значение элементу с номером n.

**listbox1.items.count** //количество элементов в списке

**combobox1.items.add('строковое выражение');**

//добавляет элемент в список

**combobox1.items.delete(4);** // удаляет 5 по счету строку

**combobox1.items.clear;** //Очищает список

**combobox1.items.loadfromfile('имя');** //загружает

данные из текстового файла

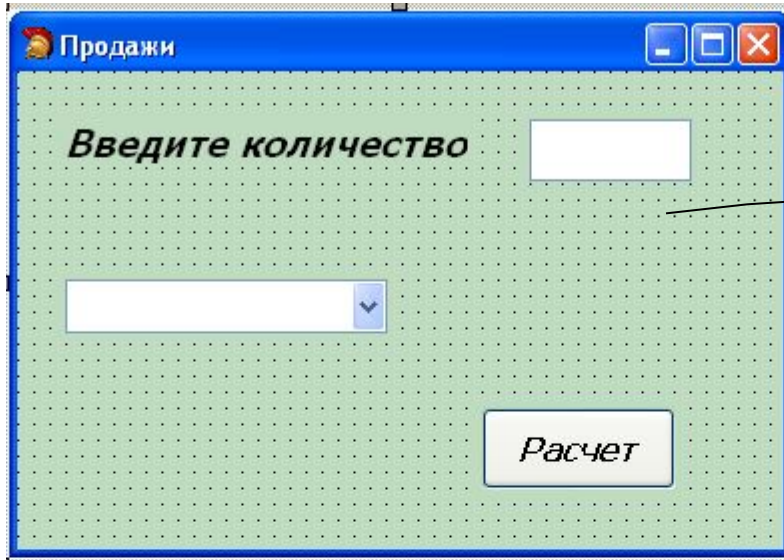
**combobox1.items.savetofile('имя');** // сохраняет

элементы списка в текстовом файле

### **Свойство только для Combobox.**

**combobox1.text** // значение элемента, который выбран в списке.

Задача: дан текстовый файл, содержащий информацию о продаже товаров: товар, количество. Вывести в список все товары, количество которых больше введенного значения в поле редактирования.

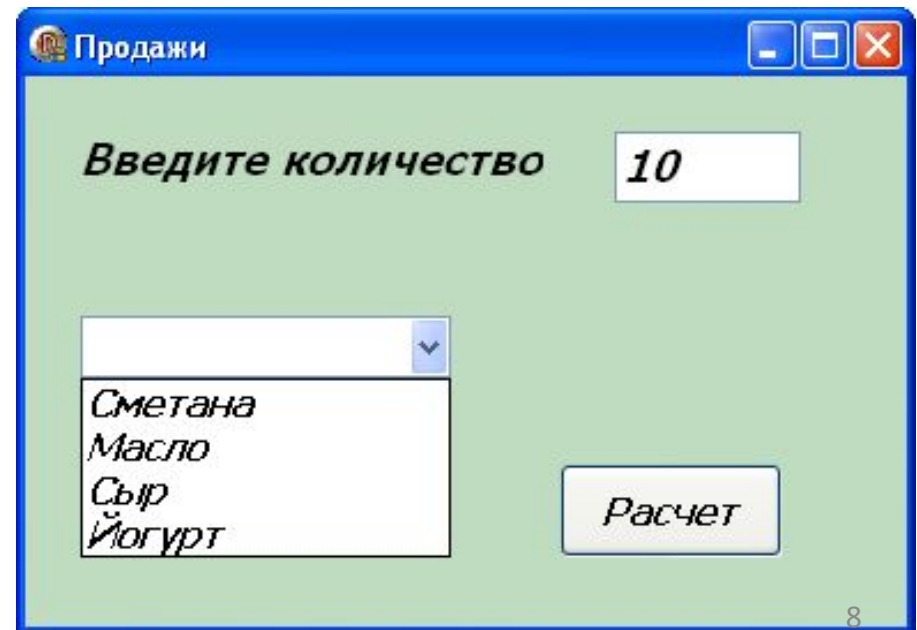


Label1

edit1

button1

combobox1





[-] **implementation**

```
{ $R *.dfm }
```

```
// Заполнение списка
```

[-] **procedure** TForm1.Button1Click(Sender: TObject);

**begin**

```
  assignfile(f, 'G:\студенты\2 курс\memo\tovar.txt');
```

```
  reset(f);
```

```
  combobox1.items.Clear;
```

```
  if edit1.text='' then begin
```

```
    showmessage('введите количество'); exit;end;
```

```
  while not eof(f) do begin
```

```
    readln(f,tov.nt);
```

```
    readln(f,tov.kol);
```

```
    if tov.kol>strtoint(edit1.Text) then combobox1.items.Add(tov.nt);
```

```
  end;
```

```
  closefile(f);
```

```
end;
```

Дан текстовый файл, содержащий информацию о сотовых телефонах: название модели, цена, изображение модели. Сформировать список из моделей телефонов. При выборе модели выводить на форму цену и изображение соответствующей модели.

Label1

Label2

Label3

Button1



ComboBox1

Image1

```
tel - Блокнот
Файл Правка Формат Вид Справка
hok ia
7654
d:\8888\1.bmp
lg
4567
d:\8888\2.bmp
sumsung
8576
d:\8888\2.bmp
```

```
price=record
  tov: string[10];
  cena: real;
  kar: string[30]; //путь к файлу с изображением телефона
end;
var
  f: textfile;
  prl:price;
```

Oncreate, onactivate

```
procedure TForm1.FormCreate(Sender: TObject);
begin
assignfile(f,'g:\студенты\2 курс\888\tel.txt');
reset(f);
combobox1.Items.Clear;
while not eof(f) do begin
  readln(f,prl.tov);
  readln(f, prl.cena);
  readln(f, prl.kar);
  combobox1.Items.Add(prl.tov);
end;
closefile(f);
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  if combobox1.Text="" then begin showmessage('Надо выбрать
товар'); exit;end;
  assignfile(f,'g:\студенты\2 курс\888\tel.txt');
  reset(f);
  while not eof(f) do begin
    readln(f,prl.tov);
    readln(f, prl.cena);
    readln(f, prl.kar);
    if trim(combobox1.text)=trim(prl.tov) then begin
      label3.Caption:=floattostr(prl.cena);
      image1.Picture.LoadFromFile(prl.kar);
    end;
  end;
  closefile(f);
end;
```

## Переключатели. Вкладка Standart



- Tcheckbox



- Tradiobutton

### Основные свойства

**Name** - Имя компонента, используется в программе для доступа к свойствам компонента. **Checkbox, radiobutton** соответственно.

**Caption** - Текст, поясняющий назначение переключателя.

**Checked** - Определяет состояние, внешний вид переключателя. Если переключатель выбран, то **checked=true**. Если переключатель не выбран, то **checked=false**.

**State** - Определяет состояние переключателя. В отличие от свойства **checked** позволяет различать выбранное, невыбранное и промежуточное состояния. Состояние переключателя определяют константы: **cbchecked (выбран)**, **cbgrayed (серый, неопределенное состояние)** и **cbunchecked (не выбран)**.

**Allowgrayed** - Определяет, может ли переключатель быть в промежуточном, неопределенном состоянии. Если **allowgrayed=false**, то переключатель может быть только выбранным или невыбранным. Если **allowgrayed=true**, то промежуточное состояние допустимо.

Например:

**If checkbox1.checked then**

**label2.caption:= ' максимальный элемент ' +inttostr(max);**

## Вкладка Win32

**TDatetimepicker** – календарь

**Date** – дата, которая выбрана.

### Функции перевода даты

var

D: Tdatetime;

S: string[10];

**Datetostr(d)** – переводит дату в строку символов;

**Strtodatetime()** – переводит строку символов в дату-время

**Strtodate()** – переводит строку символов в дату

**Now(), Date()** – текущее значение даты

**Label1.caption:=datetostr(datetimepicker1.date);**

Var

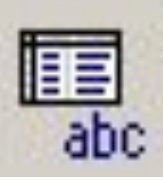
D: Tdate;

G,m,dd: word;

**Decodedate(d,g,m,dd)** – преобразует дату в год, месяц, день.

Дата должна быть записана через «.»





# Строковая таблица TStringGrid вкладка Additional

Свойство	Назначение
ColCount	Определяет число колонок таблицы
RowCount	Определяет число строк таблицы
Col	Колонка активной ячейки
Row	Строка активной ячейки
DefaultDrawing	Определяет, будет ли автоматически прорисовываться рамка (TRUE)
FixedCols	Определяет число фиксированных колонок
FixedRows	То же для строк

**Cells** - Доступ к тексту из заданной ячейки. Строки и столбцы таблицы нумеруются с нуля. Ячейка `cells[j,i]` находится на пересечении *i*-й строки и *j*-го столбца.

## Options - определяет внешний вид таблицы

goRowSizing	True	Размеры строк можно менять при работе приложения (за исключением фиксированных)
goColSizing	True	То же для колонок
goRowMoving	True	Можно переместить строку на новое место с помощью мыши
goColMoving	True	То же для колонок
goEditing	True	Можно редактировать текст (но не выделять блок!)
goTabs	True	Перемещаться по ячейкам можно через «ТАВ»
goAlwaysShowEditor	True	Таблица автоматически редактируется (иначе f2 или Enter)

// Добавление новой строки вниз таблицы.

**Begin**

**Sg.rowcount:=sg.rowcount+1;**

**Sg.rows[sg.rowcount-1].clear;**

**End;**

// Удаление текущей строки (строки, на которой стоит курсор)  
// из таблицы.

**var n: integer;**

**begin**

**if sg.rowcount=2 then exit;**

**for n:=sg.row to sg.rowcount-2 do**

**sg.rows[n]:=sg.rows[n+1];**

**sg.rowcount:=sg.rowcount-1;**

**end;**