Приклад табличного редактора в Delphi

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, ActnList, ActnCtrls, XPStyleActnCtrls, ActnMan, ToolWin,
  ActnMenus, Grids, Menus;

type
 cellr=record
  cellTEXT:string[255];
  cellWIDTH:Integer;
  cellHEIGHT:Integer;
  cellX,cellY:Integer;
  fontNAME:string[255];
  fontSIZE:Integer;
  fontCOLOR:TCOLOR;
  fontBOLD,fontITALIC,fontUNDERLINE:Boolean;
   end;

  {
   action1  - Open file
   action2  - Save file
   action3  - Save file as
   action4  - Add line
   action5  - Remove line
   action6  - Add column
   action7  - Remove column
   action8  - New file
   action9  - Font
   action10 - Bold
   action11 - Italic
   action12 - Underline
   action13 - Find dialog
   action14 - Status
  }

type
  TForm1 = class(TForm)
    ActionMainMenuBar1: TActionMainMenuBar;
    ActionManager1: TActionManager;
    ActionToolBar1: TActionToolBar;
    Action1: TAction;
    OpenDialog1: TOpenDialog;
    Action2: TAction;
    FontDialog1: TFontDialog;
    StringGrid1: TStringGrid;
```

```
    Action3: TAction;
    PopupMenu1: TPopupMenu;
    Action4: TAction;
    Action5: TAction;
    Action6: TAction;
    Action7: TAction;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    Action8: TAction;
    SaveDialog1: TSaveDialog;
    Action9: TAction;
    Action10: TAction;
    Action11: TAction;
    Action12: TAction;
    FindDialog1: TFindDialog;
    Action13: TAction;
    Action14: TAction;
    procedure StringGridUpdate; //Update StringGrid from Sheet
    procedure EmptySheet; //Erases all data in sheet
    procedure FillSheet;  //fill sheet from StringGrid
    procedure EmptyCellr(var c:cellr);  //Empty cellr structure
    procedure Action1Execute(Sender: TObject);
    procedure Action4Execute(Sender: TObject);
    procedure Action5Execute(Sender: TObject);
    procedure Action6Execute(Sender: TObject);
    procedure Action7Execute(Sender: TObject);
    procedure Action8Execute(Sender: TObject);
    procedure Action3Execute(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Action2Execute(Sender: TObject);
    procedure Action9Execute(Sender: TObject);
    procedure StringGrid1DrawCell(Sender: TObject; ACol, ARow: Integer;
      Rect: TRect; State: TGridDrawState);
    procedure Action10Execute(Sender: TObject);
    procedure Action11Execute(Sender: TObject);
    procedure Action12Execute(Sender: TObject);
    procedure FindDialog1Find(Sender: TObject);
    procedure Action13Execute(Sender: TObject);
    procedure StringGrid1SelectCell(Sender: TObject; ACol, ARow: Integer;
      var CanSelect: Boolean);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

Const
 MaxRow=250;
 MaxCol=250;
```

```
var
  Form1: TForm1;
  b:cellr;
  sheet:Array [0..MaxRow,0..MaxCol] of cellr;
  sheetH,sheetW:Integer;
  FName:String='';

implementation

{$R *.dfm}

procedure TForm1.FillSheet;
Var
 x,y:Integer;
Begin
for y:=0 to StringGrid1.RowCount-1 do
for x:=0 to StringGrid1.ColCount-1 do
 Begin
  sheet[x,y].cellTEXT:=StringGrid1.Cells[x,y];
  sheet[x,y].cellWIDTH:=StringGrid1.ColWidths[x];
  sheet[x,y].cellHeight:=StringGrid1.RowHeights[y];
 End;
End;

procedure TForm1.EmptySheet;
Var
 x,y:Integer;
Begin
For x:=0 to MaxCol do
 For y:=0 to MaxRow do
 Begin
  EmptyCellr(sheet[x,y]);
  sheet[x,y].cellX:=x;
  sheet[x,y].cellY:=y;
 End;
End;

procedure Tform1.StringGridUpdate;
Var
 x,y:Integer;
Begin
 StringGrid1.ColCount:=sheetW;
 StringGrid1.RowCount:=sheetH;
For y:=0 to sheetH-1 do
 For x:=0 to sheetW-1 do
 Begin
  stringGrid1.Cells[x,y]:=sheet[x,y].cellTEXT;
 if sheet[x,y].cellWIDTH>0 then
StringGrid1.ColWidths[x]:=sheet[x,y].cellWIDTH;
```

```
 if sheet[x,y].cellHEIGHT>0 then
StringGrid1.RowHeights[y]:=sheet[x,y].cellHEIGHT;
 End;
End;

procedure TForm1.EmptyCellr(var c:cellr);
Begin
 c.cellTEXT:='';
 c.cellWIDTH:=0;
 c.cellHEIGHT:=0;
 c.fontNAME:='';
 c.fontSIZE:=0;
 c.fontCOLOR:=0;
 c.fontBOLD:=false;
 c.fontITALIC:=false;
 c.fontUNDERLINE:=false;
End;

Procedure StringGridInsertRow(StrGrid:TStringGrid;NewRow:Integer);
Var
 Row_:Integer;
begin
 StrGrid.RowCount:=StrGrid.RowCount+1;
For Row_:=StrGrid.RowCount-1 downto NewRow do
 StrGrid.Rows[Row_].Assign(StrGrid.Rows[Row_-1]);
 StrGrid.Rows[NewRow-1].Text:='';
end;


Procedure StringGridInsertColumn(StrGrid:TStringGrid;NewColumn:Integer);
Var
 Column:Integer;
begin
 StrGrid.ColCount:=StrGrid.ColCount+1;
For Column:=StrGrid.ColCount-1 downto NewColumn do
 StrGrid.Cols[Column].Assign(StrGrid.Cols[Column-1]);
 StrGrid.Cols[NewColumn-1].Text:='';
end;

procedure StringGridDeleteColumn(var StrGrid: TStringGrid; DelColumn:
Integer);
Var Column: Integer;
begin
  If DelColumn <= StrGrid.ColCount then
  Begin
    For Column := DelColumn To StrGrid.ColCount-2 do
      StrGrid.Cols[Column].Assign(StrGrid.Cols[Column+1]);
    StrGrid.ColCount := StrGrid.ColCount-1;
  End;
end;
```

```
procedure StringGridDeleteRow(yourStringGrid: TStringGrid; ARow:
Integer);
var i: Integer;
begin
  with yourStringGrid do
  begin
    for i := ARow to RowCount-2 do
      Rows[i].Assign(Rows[i+1]);
    RowCount := RowCount - 1
  end;
end;

procedure TForm1.Action1Execute(Sender: TObject);
Var
 F:File of cellr;
 x,y,maxx,maxy:Integer;
 r:cellr;
begin
if OpenDialog1.Execute then
 Begin
  FName:=OpenDialog1.FileName;
  AssignFile(F,FName);
  Reset(F);maxx:=-2;maxy:=-2;
 While not EOF(F) do
  Begin
   Read(F,r);
   x:=r.cellX;y:=r.cellY;
  if y>maxy then maxy:=y;
  if x>maxx then maxx:=x;
   sheet[x,y]:=r;
  End;
  CloseFile(F);
  sheetW:=maxx;sheetH:=maxy;
  StringGridUpdate;
 End;
end;

procedure TForm1.Action4Execute(Sender: TObject);
begin
 StringGridInsertRow(StringGrid1,StringGrid1.Row+1);
end;

procedure TForm1.Action5Execute(Sender: TObject);
begin
 StringGridDeleteRow(StringGrid1,StringGrid1.Row);
end;

procedure TForm1.Action6Execute(Sender: TObject);
begin
 StringGridInsertColumn(StringGrid1,StringGrid1.Col+1);
end;
```

```
procedure TForm1.Action7Execute(Sender: TObject);
begin
 StringGridDeleteColumn(StringGrid1,StringGrid1.Col);
end;

procedure TForm1.Action8Execute(Sender: TObject);
begin
 EmptySheet;
 sheetW:=6;sheetH:=4;
 StringGridUpdate;
 FName:='';
end;

procedure TForm1.Action3Execute(Sender: TObject);
Var
 F:File of cellr;
 x,y:Integer;
begin
 FillSheet;
if SaveDialog1.Execute then
 Begin
  FName:=SaveDialog1.FileName;
  AssignFile(F,FName);
  Rewrite(F);
 for y:=0 to StringGrid1.RowCount do
 for x:=0 to StringGrid1.ColCount do
  Write(F,sheet[x,y]);
  CloseFile(F);
 End;
end;

procedure TForm1.FormShow(Sender: TObject);
begin
 EmptySheet;
end;

procedure TForm1.Action2Execute(Sender: TObject);
Var
 F:File of cellr;
 x,y:Integer;
begin
if FName='' then Action3Execute(Self) else
 Begin
  FillSheet;
  AssignFile(F,FName);
  Rewrite(F);
 for y:=0 to StringGrid1.RowCount do
 for x:=0 to StringGrid1.ColCount do
  Write(F,sheet[x,y]);
  CloseFile(F);
```

```
 End;
end;

procedure TForm1.Action9Execute(Sender: TObject);
Var
 x,y:Integer;
begin
If FontDialog1.Execute then
 Begin
  y:=StringGrid1.Row;x:=StringGrid1.Col;
  sheet[x,y].fontNAME:=FontDialog1.Font.Name;
  sheet[x,y].fontCOLOR:=FontDialog1.Font.Color;
  sheet[x,y].fontSIZE:=FontDialog1.Font.Size;
 if fsBold in FontDialog1.Font.Style then sheet[x,y].fontBOLD:=True
  else sheet[x,y].fontBOLD:=False;
 if fsItalic in FontDialog1.Font.Style then sheet[x,y].fontITALIC:=True
  else sheet[x,y].fontITALIC:=False;
 if fsUnderline in FontDialog1.Font.Style then
sheet[x,y].fontUNDERLINE:=True
  else sheet[x,y].fontUNDERLINE:=False;
 End;
end;

procedure TForm1.StringGrid1DrawCell(Sender: TObject; ACol, ARow:
Integer;
  Rect: TRect; State: TGridDrawState);
Var
 x,y:Integer;
begin
 StringGrid1.Canvas.Font.Style:=[];
 StringGrid1.Canvas.Font.Color:=clBlack;
 x:=ACol;y:=ARow;
if sheet[x,y].fontCOLOR<>0 then
 StringGrid1.Font.Color:=sheet[x,y].fontCOLOR;
if sheet[x,y].fontNAME<>'' then
 StringGrid1.Canvas.Font.Name:=sheet[x,y].fontNAME;
if sheet[x,y].fontSIZE<>0 then
 StringGrid1.Canvas.Font.Size:=sheet[x,y].fontSIZE;
if sheet[x,y].fontBOLD then
 StringGrid1.Canvas.Font.Style:=StringGrid1.Canvas.Font.Style+[fsBold];
if sheet[x,y].fontITALIC then
 StringGrid1.Canvas.Font.Style:=StringGrid1.Canvas.Font.Style+[fsITALIC];
if sheet[x,y].fontUNDERLINE then

StringGrid1.Canvas.Font.Style:=StringGrid1.Canvas.Font.Style+[fsUNDERLINE
];

StringGrid1.Canvas.TextRect(Rect,Rect.Left,Rect.Top,StringGrid1.Cells[x,y
]);
end;
```

```
procedure TForm1.Action10Execute(Sender: TObject);
begin
 sheet[StringGrid1.Col,StringGrid1.Row].fontBOLD:=
  not sheet[StringGrid1.Col,StringGrid1.Row].fontBOLD;
 StringGrid1.Repaint;
end;

procedure TForm1.Action11Execute(Sender: TObject);
begin
 sheet[StringGrid1.Col,StringGrid1.Row].fontITALIC:=
  not sheet[StringGrid1.Col,StringGrid1.Row].fontITALIC;
 StringGrid1.Repaint;
end;

procedure TForm1.Action12Execute(Sender: TObject);
begin
 sheet[StringGrid1.Col,StringGrid1.Row].fontUNDERLINE:=
  not sheet[StringGrid1.Col,StringGrid1.Row].fontUNDERLINE;
 StringGrid1.Repaint;
end;

procedure TForm1.FindDialog1Find(Sender: TObject);
Var
 x1,y1,x,y:Integer;
begin
 x1:=StringGrid1.Col;
 y1:=StringGrid1.Row;
for x:=x1 to SheetW do
for y:=y1 to SheetH do
 if pos(FindDialog1.FindText,StringGRid1.Cells[x,y])>0 then
  Begin
   StringGrid1.Col:=x;
   StringGrid1.Row:=y;
   break;
  End;
end;

procedure TForm1.Action13Execute(Sender: TObject);
begin
 FindDialog1.Execute;
end;

procedure TForm1.StringGrid1SelectCell(Sender: TObject; ACol,
  ARow: Integer; var CanSelect: Boolean);
begin
 Action14.Caption:='Строка: '+IntToStr(StringGrid1.Row)+
  ' Столбец : '+IntToStr(StringGrid1.Col);
end;

end.
```