Теперь я хочу показать, как создать **StringGrid**, которая позволяет работать с собой, подобно:

Grid1.Cells[1, 1].Font.Color:= clRed; Grid1.Cells[1, 2].Color:= clBlue; Grid1.Cells[1, 2].Text:= 'Это очень просто';

Реализация:

unit VesColorStringGrid; interface uses Windows, Messages, SysUtils, Classes, Controls, Grids, Graphics, Contnrs; type TVesCellProperties = class(TPersistent) private FGrid: TStringGrid; FColor: TColor; FFont: TFont; FText: string; FData: TObject; FPicture: TPicture; procedure SetColor(const Value: TColor); procedure SetFont(const Value: TFont); procedure SetText(const Value: string); procedure SetData(const Value: TObject); procedure SetPicture(const Value: TPicture); public constructor Create(AGrid: TStringGrid); virtual; property Data: TObject read FData write SetData; destructor Destroy; override; procedure Assign(Source: TPersistent); override; property Text: string read FText write SetText; property Picture: TPicture read FPicture write SetPicture; published property Color: TColor read FColor write SetColor; property Font: TFont read FFont write SetFont; end; TVesColorStringGrid = class(TStringGrid) private FPropList: TObjectList; FFixedRowParam: TVesCellProperties; FFixedColParam: TVesCellProperties; FTopLeftCorner: TVesCellProperties; function GetVesProp(ACol, ARow: Integer): TVesCellProperties; procedure SetVesProp(ACol, ARow: Integer; const Value: TVesCellProperties); procedure SetFixedColParam(const Value: TVesCellProperties); procedure SetFixedRowParam(const Value: TVesCellProperties); procedure SetTopLeftCorner(const Value: TVesCellProperties); { Private declarations } protected procedure DrawCell(ACol, ARow: Longint; ARect: TRect; AState: TGridDrawState); override; property Objects; property FixedColor; { Protected declarations } public constructor Create(Owner: TComponent); override; property Cells[ACol, ARow: Integer]: TVesCellProperties read GetVesProp write SetVesProp; { Public declarations } destructor Destroy; override; published property FixedRowParam: TVesCellProperties read FFixedRowParam write SetFixedRowParam; property FixedColParam: TVesCellProperties read FFixedColParam write SetFixedColParam; property TopLeftCorner: TVesCellProperties read FTopLeftCorner write SetTopLeftCorner; { Published declarations } end; procedure Register; implementation procedure Register; begin RegisterComponents('VesVCL', [TVesColorStringGrid]); end; { TVesCellProperties } procedure TVesCellProperties.Assign(Source: TPersistent); var vSource: TVesCellProperties; begin vSource := (Source as TVesCellProperties); inherited; FColor := vSource.FColor; FFont.Assign(vSource.FFont); FPicture.Assign(vSource.FPicture); Data := vSource.Data; end; constructor TVesCellProperties.Create; begin inherited Create; FFont := TFont.Create; FPicture := TPicture.Create; FGrid := AGrid; end; destructor TVesCellProperties.Destroy; begin FFont.Free; FPicture.Free; inherited; end; procedure TVesCellProperties.SetColor(const Value: TColor); begin FColor := Value; FGrid.Repaint;

```
 end;  procedure TVesCellProperties.SetData(const Value: TObject);  begin  FData :=
Value;  end;  procedure TVesCellProperties.SetFont(const Value: TFont);  begin
FFont.Assign(Value);  FGrid.Repaint;  end;  procedure TVesCellProperties.SetPicture(const
Value: TPicture);  begin  FPicture.Assign(Value);  FGrid.Repaint;  end;  procedure
TVesCellProperties.SetText(const Value: string);  begin  FText := Value;  FGrid.Repaint;
end;  { TVesColorStringGrid }  constructor TVesColorStringGrid.Create(Owner:
TComponent);  begin  inherited;  FPropList := TObjectList.Create(True);
FFixedRowParam := TVesCellProperties.Create(Self);  FFixedColParam :=
TVesCellProperties.Create(Self);  FTopLeftCorner := TVesCellProperties.Create(Self);
with FFixedRowParam do  begin  Color := FixedColor;  Font := Self.Font;  end;
with FFixedColParam do  begin  Color := FixedColor;  Font := Self.Font;  end;  with
FTopLeftCorner do  begin  Color := FixedColor;  Font := Self.Font;  end;  end;
destructor TVesColorStringGrid.Destroy;  begin  FPropList.Free;  inherited;  end;
procedure TVesColorStringGrid.DrawCell(ACol,  ARow: Integer; ARect: TRect;  AState:
TGridDrawState);  var  VParam: TVesCellProperties;  begin  inherited;  with Canvas do
 begin  if not (gdFixed in AState) then  VParam := Cells[ACol, ARow]  else if ACol >
FixedRows - 1 then  VParam := FixedColParam  else if ARow > FixedCols - 1 then
VParam := FFixedRowParam  else  VParam := FTopLeftCorner;  Brush.Color :=
VParam.Color;  Font.Assign(Vparam.FFont);  FillRect(ARect);  TextRect(ARect,
ARect.Left + 2, ARect.Top + 2,  Cells[ACol,ARow].Text);  end;  end;  function
TVesColorStringGrid.GetVesProp(ACol,  ARow: Integer): TVesCellProperties;  var  VProp:
TVesCellProperties;  begin  if Assigned(Objects[ACol, ARow]) then  Result :=
TVesCellProperties(Objects[ACol, ARow])  else  begin  VProp :=
TVesCellProperties.Create(Self);  with VProp do  begin  FColor := Self.Color;
Font.Assign(Font);  end;  FPropList.Add(VProp);  Objects[ACol, ARow] := VProp;
Result := VProp;  end;  end;  procedure TVesColorStringGrid.SetFixedColParam(  const
Value: TVesCellProperties);  begin  FFixedColParam.Assign(Value);  end;  procedure
TVesColorStringGrid.SetFixedRowParam(  const Value: TVesCellProperties);  begin
FFixedRowParam.Assign(Value);  end;  procedure TVesColorStringGrid.SetTopLeftCorner(
 const Value: TVesCellProperties);  begin  FTopLeftCorner.Assign(Value);  end;  procedure
TVesColorStringGrid.SetVesProp(ACol,  ARow: Integer; const Value: TVesCellProperties);
begin  GetVesProp(ACol, ARow);  TVesCellProperties(Objects[ACol, ARow]).Assign(Value);
 end;
```

По материалам http://delphi.3000.com