```
unit Unit1;
interface
uses
 Winapi.Windows, Winapi.Messages, Sys-
tem.SysUtils, System.Variants, System.Classes,
Vcl.Graphics,
Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Im-
aging.jpeg, Vcl.ExtCtrls,
Vcl.StdCtrls, Vcl.Buttons, Vcl.Imag-
ing.pngimage;
type
TForm2 = class(TForm)
Image1:TImage;
Label1:TLabel;
SpeedButton1: TSpeedButton;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
procedure FormShow(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure SpeedButton1Click(Sender:
TObject);
 procedure SpeedButton2Click(Sender:
TObject);
 procedure SpeedButton3Click(Sender:
TObject);
 private
 FForm3Shown:Boolean;
 { Private declarations }
 public
 { Public declarations }
 end;
var
 Form2: TForm2;
implementation
{$R *.dfm}
uses Unit2, Unit3, Unit4;
procedure TForm2.FormCreate(Sender:
TObject);
begin
SpeedButton1.Font.Name := 'Ink Free';
SpeedButton3.Font.Name := 'Ink Free';
SpeedButton2.Font.Name := 'Ink Free'
end;
procedure TForm2.FormShow(Sender:
TObject);
begin
 if not FForm3Shown then
 begin
 Form3.ShowModal;
 FForm3Shown := True;
 end;
end;
procedure TForm2.SpeedButton1Click(Sender:
TObject);
begin
Form2.Hide;
Form5.Show;
end;
procedure TForm2.SpeedButton2Click(Sender:
TObject);
begin
Close;
end;
procedure TForm2.SpeedButton3Click(Sender:
TObject);
begin
ShowMessage('Разработала учащаяся группы
ПЗТ-41 Рубис Диана' + #13 + 'Курсовое про-
ект: игровое приложение "Нонограммы"');
end;
end.
unit Unit2;
interface
uses
 Winapi.Windows, Winapi.Messages, Sys-
tem.SysUtils, System.Variants, System.Classes,
Vcl.Graphics,
 Vcl.Controls, Vcl.Forms, Vcl.Dialogs,
Vcl.StdCtrls, Vcl.ExtCtrls, Vcl.ComCtrls,
 Vcl.Imaging.jpeg;
type
 TForm3 = class(TForm)
 Image1: TImage;
 Label1: TLabel;
 Timer1: TTimer;
 ProgressBar1: TProgressBar;
 procedure Timer1Timer(Sender: TObject);
 private
 FFullText: string;
 FCurrentIndex: Integer;
 public
 { Private declarations }
 public
 { Public declarations }
 end;
var
 Form3: TForm3;
implementation
{$R *.dfm}
uses Unit1;
procedure TForm3.Timer1Timer(Sender:
TObject);
```

```
begin
 // Увеличиваем значение ProgressBar
 ProgressBar1.Position := ProgressBar1.Posi-
tion + 15;
 Sleep(10);
 // Проверка заполнен ли ProgressBar
 if ProgressBar1.Position >= 100 then
 begin
 Timer1.Enabled := False; // Останавливаем
таймер
 Form2.Show;
 Self.Close;
 end;
end;
end. Form2.Show; // Показываем Form3
 Self.Close; // Закрываем текущую форму
 end;
end;
end.
unit Unit3;
interface
uses
 Winapi.Windows, Winapi.Messages, Sys-
tem.SysUtils, System.Variants, System.Classes,
Vcl.Graphics,
 Vcl.Controls, Vcl.Forms, Vcl.Dialogs,
Vcl.StdCtrls, Vcl.Grids, Vcl.Buttons,
 Vcl.ExtCtrls, Vcl.Menus, ShellApi;
type
 TForm4 = class(TForm)
 StringGrid1: TStringGrid;
 SpeedButton1: TSpeedButton;
 SpeedButton3: TSpeedButton;
 Image1: TImage;
 SpeedButton2: TSpeedButton;
 MainMenu1: TMainMenu;
 N1: TMenuItem;
 procedure InitializeSolution;
 procedure FormCreate(Sender: TObject);
 procedure StringGrid1MouseDown(Sender:
TObject; Button: TMouseButton;
 Shift: TShiftState; X, Y: Integer);
 procedure StringGrid1DrawCell(Sender:
TObject; ACol, ARow: Integer;
 Rect: TRect; State: TGridDrawState);
 procedure LoadLevelFromFile(const Col-
umnsFilePath, RowsFilePath: string; const
Grid: TStringGrid);
 procedure SetFilePathAndInitializeSolu-
tion(const FilePath: string);
 procedure SpeedButton1Click(Sender:
TObject);
 procedure SpeedButton3Click(Sender:
TObject);
 procedure SpeedButton2Click(Sender:
TObject);
 private
 SolutionGrid: array[1..10, 1..10] of Integer;
 public
 LevelsCompleted: Integer;
 FilePath: string; // Путь к файлу
 GridPath: string; // Путь к файлу с заголов-
ками строк и столбцов
 end;
var
 Form4: TForm4;
 LevelsCompleted: Integer = 0;
implementation
{$R *.dfm}
uses Unit4, Unit1;
procedure TForm4.SetFilePathAndInitializeSo-
lution(const FilePath: string);
begin
 Self.FilePath := FilePath;
 InitializeSolution;
end;
procedure TForm4.SpeedButton1Click(Sender:
TObject);
var
 ACol, ARow: Integer;
 IsCorrect: Boolean;
begin
 InitializeSolution;
 IsCorrect := True;
 for ACol := 1 to StringGrid1.ColCount - 1 do
 begin
 for ARow := 1 to StringGrid1.RowCount - 1
do
 begin
 if ((StringGrid1.Cells[ACol, ARow] = '1') and
(SolutionGrid[ACol, ARow] <> 1)) or
 ((StringGrid1.Cells[ACol, ARow] = '') and
(SolutionGrid[ACol, ARow] = 1)) then
 begin
 IsCorrect := False;
 Break;
 end;
 end;
 if not IsCorrect then
 Break;
 end;
```

```
 if IsCorrect then
 begin
 ShowMessage('Правильное решение!');
 Inc(LevelsCompleted);
 if not Assigned(Form5) then
 Form5 := TForm5.Create(Self);
 Form5.UnlockLevels;
 end
 else
 begin
 ShowMessage('Неправильное решение!');
 end;
end;
procedure TForm4.SpeedButton2Click(Sender:
TObject);
begin
Form4.Hide;
Form2.Show;
end;
procedure TForm4.SpeedButton3Click(Sender:
TObject);
var
 i, j: Integer;
begin
 // Очистка таблицы
 for i := 1 to StringGrid1.ColCount - 1 do
 begin
 for j := 1 to StringGrid1.RowCount - 1 do
 begin
 StringGrid1.Objects[i, j] := TObject(clWhite);
 StringGrid1.Cells[i, j] := '';
 end;
 end;
 // обновление таблицы
 StringGrid1.Invalidate;
 Form4.Hide;
 Form5.Show;
end;
procedure TForm4.FormCreate(Sender:
TObject);
var
 i, j: Integer;
begin
 SpeedButton1.Font.Name := 'Ink Free';
 SpeedButton2.Font.Name := 'Ink Free';
 SpeedButton3.Font.Name := 'Ink Free';
 StringGrid1.Options := StringGrid1.Options +
[goFixedVertLine, goFixedHorzLine, goVert-
Line, goHorzLine, goRangeSelect, goDrawFo-
cusSelected];
 StringGrid1.Options := StringGrid1.Options -
[goEditing];
 for i := 0 to StringGrid1.ColCount - 1 do
 for j := 0 to StringGrid1.RowCount - 1 do
 begin
 StringGrid1.Objects[i, j] := TObject(clWhite);
 end;
end;
procedure TForm4.InitializeSolution;
var
 F: TextFile;
 ACol, ARow, number: Integer;
begin
 if FilePath = '' then
 begin
 ShowMessage('Не указан путь к файлу дан-
ных.');
 Exit;
 end;
 AssignFile(F, FilePath);
 try
 Reset(F);
 for ACol := 1 to StringGrid1.ColCount - 1 do
 begin
 for ARow := 1 to StringGrid1.RowCount - 1
do
 begin
 if not Eof(F) then
 begin
 Read(F, number);
 SolutionGrid[ACol, ARow] := number;
 end
 else
 begin
 ShowMessage('Недостаточно данных в
файле.');
 Exit;
 end;
 end;
 Readln(F);
 end;
 CloseFile(F);
 except
 on E: Exception do
 begin
 ShowMessage('Ошибка чтения файла: ' +
E.Message);
 CloseFile(F);
 end;
 end;
end;
```

```
procedure TForm4.StringGrid1Draw-
Cell(Sender: TObject; ACol, ARow: Integer;
 Rect: TRect; State: TGridDrawState);
begin
 // Установка цвета для закрашивания ячеек
 StringGrid1.Canvas.Brush.Color :=
TColor(StringGrid1.Objects[ACol, ARow]);
 StringGrid1.Canvas.FillRect(Rect);
 StringGrid1.Canvas.TextRect(Rect, Rect.Left
+ 2, Rect.Top + 2, StringGrid1.Cells[ACol,
ARow]);
 // Отрисовка крестика, если цвет красный
 if TColor(StringGrid1.Objects[ACol, ARow])
= clRed then
 begin
 StringGrid1.Canvas.Pen.Color := clBlack;
 StringGrid1.Canvas.MoveTo(Rect.Left,
Rect.Top);
 StringGrid1.Canvas.LineTo(Rect.Right,
Rect.Bottom);
 StringGrid1.Canvas.MoveTo(Rect.Left,
Rect.Bottom);
 StringGrid1.Canvas.LineTo(Rect.Right,
Rect.Top);
 end;
end;
procedure TForm4.String-
Grid1MouseDown(Sender: TObject; Button:
TMouseButton;
 Shift: TShiftState; X, Y: Integer);
var
 ACol, ARow: Integer;
 CurrentColor: TColor;
begin
 StringGrid1.MouseToCell(X, Y, ACol,
ARow);
 if (ACol > 0) and (ARow > 0) then
 begin
 CurrentColor := TColor(StringGrid1.Ob-
jects[ACol, ARow]);
 if CurrentColor = clWhite then
 begin
 SolutionGrid[ACol, ARow] := 1;
 StringGrid1.Objects[ACol, ARow] :=
TObject(clBlack);
 StringGrid1.Cells[ACol, ARow] := '1';
 end
 else if CurrentColor = clBlack then
 begin
 SolutionGrid[ACol, ARow] := 0;
 StringGrid1.Objects[ACol, ARow] :=
TObject(clRed);
 StringGrid1.Cells[ACol, ARow] := '';
 end
 else
 begin
 SolutionGrid[ACol, ARow] := 0;
 StringGrid1.Objects[ACol, ARow] :=
TObject(clWhite);
 StringGrid1.Cells[ACol, ARow] := '';
 end;
 StringGrid1.Invalidate;
 end;
end;
function Min(const A, B: Integer): Integer;
begin
 if A < B then
 Result := A
 else
 Result := B;
end;
procedure TForm4.LoadLevelFromFile(const
ColumnsFilePath, RowsFilePath: string; const
Grid: TStringGrid);
var
 F: TextFile;
 RowIndex, ColIndex: Integer;
 Line: string;
 Numbers: TStringList;
begin
 // Проверка существования файлов
 if not FileExists(ColumnsFilePath) then
 begin
 ShowMessage('Файл для столбцов не
найден: ' + ColumnsFilePath);
 Exit;
 end;
 if not FileExists(RowsFilePath) then
 begin
 ShowMessage('Файл для строк не найден: ' +
RowsFilePath);
 Exit;
 end;
 // Заполнение данных для столбцов
 AssignFile(F, ColumnsFilePath);
 Reset(F);
 ColIndex := 1;
 try
 while not Eof(F) do
 begin
 Readln(F, Line);
```

```
 Numbers := TStringList.Create;
 try
 Numbers.Delimiter := ' ';
 Numbers.DelimitedText := Line;
 for RowIndex := 0 to Min(Numbers.Count - 1,
Grid.RowCount - 1) do
 Grid.Cells[ColIndex, RowIndex] := Num-
bers[RowIndex];
 finally
 Numbers.Free;
 end;
 Inc(ColIndex);
 if ColIndex >= Grid.ColCount then
 Break;
 end;
 finally
 CloseFile(F);
 end;
 // Заполнение данных для строк
 AssignFile(F, RowsFilePath);
 Reset(F);
 RowIndex := 1;
 try
 while not Eof(F) do
 begin
 Readln(F, Line);
 Numbers := TStringList.Create;
 try
 Numbers.Delimiter := ' ';
 Numbers.DelimitedText := Line;
 for ColIndex := 0 to Min(Numbers.Count - 1,
Grid.ColCount - 1) do
 Grid.Cells[ColIndex, RowIndex] := Num-
bers[ColIndex];
 finally
 Numbers.Free;
 end;
 Inc(RowIndex);
 if RowIndex >= Grid.RowCount then
 Break;
 end;
 finally
 CloseFile(F);
 end;
end;
procedure TForm4.N1Click(Sender: TObject);
begin
ShellExecute(0, 'open', PChar ('help.chm'), nil,
nil, SW_SHOW);
end;
end.
```

```
unit Unit4;
interface
uses
 Winapi.Windows, Winapi.Messages, Sys-
tem.SysUtils, System.Variants, System.Classes,
Vcl.Graphics,
 Vcl.Controls, Vcl.Forms, Vcl.Dialogs,
Vcl.StdCtrls, Vcl.Buttons, Vcl.ExtCtrls,
 Vcl.Imaging.pngimage;
type
 TForm5 = class(TForm)
 GroupBox1: TGroupBox;
 GroupBox2: TGroupBox;
 GroupBox3: TGroupBox;
 rbEasyLevel1: TRadioButton;
 rbEasyLevel2: TRadioButton;
 rbEasyLevel3: TRadioButton;
 rbMediumLevel1: TRadioButton;
 rbMediumLevel2: TRadioButton;
 rbMediumLevel3: TRadioButton;
 rbHardLevel1: TRadioButton;
 rbHardLevel2: TRadioButton;
 rbHardLevel3: TRadioButton;
 Image1: TImage;
 procedure FormCreate(Sender: TObject);
 procedure RadioButtonClick(Sender:
TObject);
 procedure LoadLevelFile(const FileName:
string);
 procedure FormShow(Sender: TObject);
 private
 FileName: string;
 { Private declarations }
 public
 FilePath:string;
 procedure UnlockLevels;
 end;
var
 Form5: TForm5;
 LevelsCompleted: Integer = 0;
implementation
{$R *.dfm}
uses Unit1, Unit3;
procedure TForm5.FormCreate(Sender:
TObject);
begin
GroupBox1.Font.Name := 'Ink Free';
GroupBox2.Font.Name := 'Ink Free';
GroupBox3.Font.Name := 'Ink Free';
 rbEasyLevel1.OnClick := RadioButtonClick;
 rbEasyLevel2.OnClick := RadioButtonClick;
```

```
 rbEasyLevel3.OnClick := RadioButtonClick;
 rbMediumLevel1.OnClick := RadioButton-
Click;
 rbMediumLevel2.OnClick := RadioButton-
Click;
 rbMediumLevel3.OnClick := RadioButton-
Click;
 rbHardLevel1.OnClick := RadioButtonClick;
 rbHardLevel2.OnClick := RadioButtonClick;
 rbHardLevel3.OnClick := RadioButtonClick;
end;
procedure TForm5.FormShow(Sender:
TObject);
begin
 rbEasyLevel1.Checked := False;
 rbEasyLevel2.Checked := False;
 rbEasyLevel3.Checked := False;
 rbMediumLevel1.Checked := False;
 rbMediumLevel2.Checked := False;
 rbMediumLevel3.Checked := False;
 rbHardLevel1.Checked := False;
 rbHardLevel2.Checked := False;
 rbHardLevel3.Checked := False;
end;
procedure TForm5.UnlockLevels;
begin
 if Form4.LevelsCompleted >= 1 then
 begin
 rbEasyLevel1.Enabled := False;
 end;
 if Form4.LevelsCompleted >= 2 then
 begin
 rbEasyLevel2.Enabled := False;
 end;
 if Form4.LevelsCompleted >= 3 then
 begin
 GroupBox2.Enabled := True;
 GroupBox2.Visible := True;
 rbEasyLevel3.Enabled := False;
 GroupBox2.Repaint;
 end;
 if Form4.LevelsCompleted >= 4 then
 begin
 rbMediumLevel1.Enabled := False;
 end;
 if Form4.LevelsCompleted >= 5 then
 begin
 rbMediumLevel2.Enabled := False;
 end;
 if Form4.LevelsCompleted >= 6 then
 begin
 GroupBox3.Enabled := True;
 GroupBox3.Visible := True;
 rbMediumLevel3.Enabled := False;
 GroupBox3.Repaint;
 end;
 if Form4.LevelsCompleted >= 7 then
 begin
 rbHardLevel1.Enabled := False;
 end;
 if Form4.LevelsCompleted >= 8 then
 begin
 rbHardLevel2.Enabled := False;
 end;
 if Form4.LevelsCompleted >= 9 then
 begin
 rbHardLevel3.Enabled := False;
 end;
end;
procedure TForm5.RadioButtonClick(Sender:
TObject);
var
 FileName: string;
begin
 if Sender = rbEasyLevel1 then
 FileName := 'EasyLevel1'
 else if Sender = rbEasyLevel2 then
 FileName := 'EasyLevel2'
 else if Sender = rbEasyLevel3 then
 FileName := 'EasyLevel3'
 else if Sender = rbMediumLevel1 then
 FileName := 'MediumLevel1'
 else if Sender = rbMediumLevel2 then
 FileName := 'MediumLevel2'
 else if Sender = rbMediumLevel3 then
 FileName := 'MediumLevel3'
 else if Sender = rbHardLevel1 then
 FileName := 'HardLevel1'
 else if Sender = rbHardLevel2 then
 FileName := 'HardLevel2'
 else if Sender = rbHardLevel3 then
 FileName := 'HardLevel3';
 LoadLevelFile(FileName);
end;
procedure TForm5.LoadLevelFile(const File-
Name: string);
var
 LevelsPath, FullPath, FullPathC, FullPathR:
string;
begin
 LevelsPath := ExtractFilePath(Application.Ex-
eName);
```

```
 LevelsPath := IncludeTrailingPathDelim-
iter(LevelsPath) + 'levels\';
 FullPath := LevelsPath + FileName + '\resh-
enie.txt';
 FullPathC := LevelsPath + FileName +
'\gridC.txt';
 FullPathR := LevelsPath + FileName +
'\gridR.txt';
 if not FileExists(FullPath) then
 begin
 ShowMessage('Файл не найден: ' + FullPath);
 Exit;
 end;
 if not Assigned(Form4) then
 Form4 := TForm4.Create(Application);
 Form4.FilePath := FullPath; // Передача пути
к файлу
 Form4.LoadLevelFromFile(FullPathC, Full-
PathR, Form4.StringGrid1); // Передача путей
к файлам для столбцов и строк
 Form4.Show;
 Form5.Hide;
end;
end.
```