**Гладкий Максим 851001 Контрольная работа 20.03.2019 Вариант 42**

**Задание:**Дан двухмерный массив. Упорядочить его элементы в порядке возрастания. **Код Delphi:**

**unit** UnitMain;

**interface**

**uses**

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Menus, Vcl.StdCtrls, Vcl.Grids;

**type**

TMatrix = **array**[1..5] **of array**[1..5] **of** Integer;

TArr = **array**[1..25] **of** Integer;

TMain = **class**(TForm)

SG: TStringGrid;

SGR: TStringGrid;

SortBtn: TButton;

RowsAm: TEdit;

ColumnsAm: TEdit;

Text1: TLabel;

Text2: TLabel;

CheckBtn: TButton;

Popup: TPopupMenu;

**procedure** RowsAmKeyPress(Sender: TObject; **var** Key: Char);

**procedure** CheckBtnClick(Sender: TObject);

**procedure** SortBtnClick(Sender: TObject);

**procedure** SGKeyPress(Sender: TObject; **var** Key: Char);

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**var**

Main: TMain;

**implementation**

{$R \*.dfm}

**procedure** TMain.CheckBtnClick(Sender: TObject);

**begin**

**if** (Length(RowsAm.Text) > 0)**and**(Length(ColumnsAm.Text) > 0) **then**

**begin**

SG.Enabled := true;

SG.RowCount := StrToInt(RowsAm.Text);

SG.ColCount := StrToInt(ColumnsAm.Text);

SGR.RowCount := StrToInt(RowsAm.Text);

SGR.ColCount := StrToInt(ColumnsAm.Text);

SortBtn.Enabled := true;

**end**

**else**

**begin**

ShowMessage('Error. Enter numbers [2..5] in blank fields.');

CheckBtn.Enabled := false;

SG.Enabled := false;

SortBtn.Enabled := false;

SGR.Visible := false;

**end**;

**end**;

**procedure** TMain.RowsAmKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **set of** char;

**begin**

Numerals := ['2'..'5', #8];

CheckBtn.Enabled := true;

SortBtn.Enabled := false;

SG.Enabled := false;

SGR.Visible := false;

**with** Sender **as** TEdit **do**

**begin**

**if not** (Key **in** Numerals) **then**

Key := #0;

**if** (Length(Text) = 1) **and** (Key <> #8) **then**

Key := #0;

**end**;

**end**;

**procedure** Sorting(**var** Matrix: TMatrix; Sender: TObject);

**var**

i, j, k, Buffer: Integer;

Buff: TArr;

**begin**

k := 1;

**for** i := 1 **to** Main.SG.RowCount **do**

**for** j := 1 **to** Main.SG.ColCount **do**

**begin**

Buff[k] := Matrix[i, j];

inc(k);

**end**;

**for** i := 1 **to** k - 1 **do**

**for** j := i **to** k **do**

**if** Buff[i] > Buff[j] **then**

**begin**

Buffer := Buff[i];

Buff[i] := Buff[j];

Buff[j] := Buffer;

**end**;

k := 1;

**for** i := 1 **to** Main.SG.RowCount **do**

**for** j := 1 **to** Main.SG.ColCount **do**

**begin**

Matrix[i, j] := Buff[k];

inc(k);

**end**;

**end**;

**procedure** TMain.SGKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **set of** char;

**begin**

SGR.Visible := false;

Numerals := ['0'..'9', #8, '-'];

**if not** (Key **in** Numerals) **then**

Key := #0;

**if** (Length(SG.Cells[SG.Col, SG.Row]) = 2) **and** (Key <> #8) **then**

Key := #0;

**if** (SG.Cells[SG.Col, SG.Row] = '0') **and** (Key <> #8) **then**

Key := #0;

**end**;

**procedure** TMain.SortBtnClick(Sender: TObject);

**var**

i, j: Integer;

Matrix: TMatrix;

**begin**

**for** i := 0 **to** SG.RowCount - 1 **do**

**for** j := 0 **to** SG.ColCount - 1 **do**

Matrix[i + 1, j + 1] := StrtoInt(SG.Cells[j, i]);

Sorting(Matrix, Sender);

**for** i := 0 **to** SG.RowCount - 1 **do**

**for** j := 0 **to** SG.ColCount - 1 **do**

SGR.Cells[j, i] := InttoStr(Matrix[i + 1, j + 1]);

SGR.Visible := true;

**end**;

**end**.