unit MainFormUnit;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Buttons, Vcl.ExtCtrls,

Vcl.Menus, Vcl.ToolWin, Vcl.ActnMan, Vcl.ActnCtrls, Vcl.ActnMenus,

Vcl.Imaging.pngimage, ShellAPI;

type

TMainForm = class(TForm)

StartBtn: TPanel;

ManualBtn: TPanel;

ExitBtn: TPanel;

Background: TImage;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

StartBtnImg: TImage;

ManualBtnImg: TImage;

ExitBtnImg: TImage;

Image5: TImage;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N6: TMenuItem;

procedure StartBtnClick(Sender: TObject);

procedure ExitBtnClick(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure N6Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure FormDestroy(Sender: TObject);

procedure ManualBtnClick(Sender: TObject);

procedure N2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

MainForm: TMainForm;

implementation

{$R \*.dfm}

uses SplashFormUnit, SectionsFormUnit, ArrayFormUnit, CodeFormUnit, TheoryFormUnit;

procedure TMainForm.FormCreate(Sender: TObject);//загрузка шрифтов

var filepath:string;

begin

filepath:=extractfilepath(application.ExeName);

addfontresource(PChar(filepath + 'font\days.ttf'));

end;

procedure TMainForm.FormDestroy(Sender: TObject);//выгрузка шрифтов

var filepath:string;

begin

filepath:=extractfilepath(application.ExeName);

removefontresource(PChar(filepath + 'font\days.ttf'));

end;

procedure TMainForm.ManualBtnClick(Sender: TObject);

begin

TheoryForm.SetTheoryID(0);

TheoryForm.show;

end;

procedure TMainForm.N2Click(Sender: TObject);

begin

ShellExecute(0,PChar('Open'),PChar('manual.chm'),nil,nil,SW\_SHOW);

end;

procedure TMainForm.N3Click(Sender: TObject);

begin

if not Assigned(ArrayForm) then

ArrayForm:= TArrayForm.Create(Application); // Создать форму, если она еще не создана

ArrayForm.SetMenuID(1); // Установить Menu\_ID

ArrayForm.show;

MainForm.hide;

end;

procedure TMainForm.N4Click(Sender: TObject);

begin

if not Assigned(ArrayForm) then

ArrayForm:= TArrayForm.Create(Application); // Создать форму, если она еще не создана

ArrayForm.SetMenuID(2); // Установить Menu\_ID

ArrayForm.show;

MainForm.hide;

end;

procedure TMainForm.N5Click(Sender: TObject);

begin

if not Assigned(ArrayForm) then

ArrayForm:= TArrayForm.Create(Application); // Создать форму, если она еще не создана

ArrayForm.SetMenuID(3);

ArrayForm.show;

MainForm.hide;

end;

procedure TMainForm.N6Click(Sender: TObject);

begin

if not Assigned(ArrayForm) then

ArrayForm:= TArrayForm.Create(Application); // Создать форму, если она еще не создана

ArrayForm.SetMenuID(4);

ArrayForm.show;

MainForm.hide;

end;

procedure TMainForm.StartBtnClick(Sender: TObject);

begin

SectionsForm.show();

MainForm.hide();

end;

procedure TMainForm.ExitBtnClick(Sender: TObject);

begin

close;

end;

end.

unit SectionsFormUnit;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Buttons, Vcl.ExtCtrls,

Vcl.Menus, Vcl.Imaging.pngimage;

type

TSectionsForm = class(TForm)

CreateArr\_button: TPanel;

SortArr\_Button: TPanel;

FindArr\_Button: TPanel;

EditArr\_Button: TPanel;

BackToHome\_Button: TPanel;

Image1: TImage;

Image2: TImage;

Image3: TImage;

Image4: TImage;

Image5: TImage;

Image6: TImage;

procedure CreateArr\_buttonClick(Sender: TObject);

procedure BackToHome\_ButtonClick(Sender: TObject);

procedure SortArr\_ButtonClick(Sender: TObject);

procedure FindArr\_ButtonClick(Sender: TObject);

procedure EditArr\_ButtonClick(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure CreateParams(var Params: TCreateParams) ; override;

private

{ Private declarations }

public

{ Public declarations }

end;

var

SectionsForm: TSectionsForm;

implementation

{$R \*.dfm}

uses MainFormUnit, ArrayFormUnit, TheoryFormUnit;

procedure TSectionsForm.BackToHome\_ButtonClick(Sender: TObject);

begin

MainForm.show;

SectionsForm.close;

end;

procedure TSectionsForm.CreateArr\_buttonClick(Sender: TObject);

begin

if not Assigned(ArrayForm) then

ArrayForm:= TarrayForm.Create(Application); // Создать форму, если она еще не создана

ArrayForm.SetMenuID(1); //установил Menu\_ID

TheoryForm.SetTheoryID(1);

ArrayForm.show;

SectionsForm.hide;

end;

procedure TSectionsForm.EditArr\_ButtonClick(Sender: TObject);

begin

if not Assigned(ArrayForm) then

ArrayForm:= TArrayForm.Create(Application); // Создать форму, если она еще не создана

ArrayForm.SetMenuID(4);

TheoryForm.SetTheoryID(4);

ArrayForm.show;

SectionsForm.hide;

end;

procedure TSectionsForm.FindArr\_ButtonClick(Sender: TObject);

begin

if not Assigned(ArrayForm) then

ArrayForm:= TArrayForm.Create(Application); // Создать форму, если она еще не создана

ArrayForm.SetMenuID(3);

TheoryForm.SetTheoryID(3);

ArrayForm.show;

SectionsForm.hide;

end;

procedure TSectionsForm.FormClose(Sender: TObject; var Action: TCloseAction);

begin

mainform.Show();

end;

procedure TSectionsForm.SortArr\_ButtonClick(Sender: TObject);

begin

if not Assigned(ArrayForm) then

ArrayForm:= TArrayForm.Create(Application); // Создать форму, если она еще не создана

ArrayForm.SetMenuID(2); //установил Menu\_ID

TheoryForm.SetTheoryID(2);

ArrayForm.show;

SectionsForm.hide;

end;

procedure TSectionsForm.CreateParams(var Params: TCreateParams) ;//процедура для того чтобы ярлык в панели задач не пропадал

begin

inherited;

Params.ExStyle := Params.ExStyle or WS\_EX\_APPWINDOW;

Params.WndParent := 0;

end;

end.

unit SplashFormUnit;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ComCtrls, Vcl.ExtCtrls, CommCtrl,

GIFImg, Vcl.Imaging.pngimage;

type

TSplashForm = class(TForm)

Timer1: TTimer;

Image1: TImage;

Image2: TImage;

{procedure Button1Click(Sender: TObject);}

procedure Timer1Timer(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

SplashForm: TSplashForm;

implementation

{$R \*.dfm}

uses MainFormUnit;

procedure TSplashForm.FormCreate(Sender: TObject);

begin

(Image1.Picture.Graphic as TGIFImage).Animate := True;

end;

procedure TSplashForm.Timer1Timer(Sender: TObject);

begin

timer1.enabled:=false;

end;

end.

unit TheoryFormUnit;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.OleCtrls, SHDocVw,

Vcl.ExtCtrls, Vcl.Imaging.pngimage;

type

TTheoryForm = class(TForm)

Exit\_button: TPanel;

WebBrowser1: TWebBrowser;

Name\_Label: TLabel;

Image1: TImage;

Image2: TImage;

procedure FormShow(Sender: TObject);

procedure SetTheoryID(NewTheoryID: Integer);

procedure Exit\_buttonClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

TheoryForm: TTheoryForm;

Theory\_ID:integer;

implementation

{$R \*.dfm}

uses ArrayFormUnit, SectionsFormUnit;

procedure TTheoryForm.SetTheoryID(NewTheoryID: Integer);

begin

Theory\_ID := NewTheoryID;

end;

procedure TTheoryForm.Exit\_buttonClick(Sender: TObject);

begin

TheoryForm.Close;

end;

procedure TTheoryForm.FormShow(Sender: TObject);

var

filepath:string;

begin

filepath:=extractfilepath(application.ExeName);

if Theory\_ID=0 then

begin

Name\_Label.Caption:='Динамические массивы';

webbrowser1.navigate(PChar(filepath + 'theory\Динамические массивы.htm'));

end

else

if Theory\_ID=1 then

begin

Name\_Label.Caption:='Создание массива';

webbrowser1.navigate(PChar(filepath + 'theory\Создание массива.htm'));

end

else

if Theory\_ID=2 then

begin

Name\_Label.Caption:='Сортировка массива';

webbrowser1.navigate(PChar(filepath + 'theory\Сортировка массива.htm'));

end

else

if Theory\_ID=3 then

begin

Name\_Label.Caption:='Поиск в массиве';

webbrowser1.navigate(PChar(filepath + 'theory\Поиск в массиве.htm'));

end

else

if Theory\_ID=4 then

begin

Name\_Label.Caption:='Чтение и изменение элементов';

webbrowser1.navigate(PChar(filepath + 'theory\Чтение и изменение элементов.htm'));

end;

end;

end.

unit ArrayFormUnit;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Buttons, Vcl.Grids,

Vcl.ExtCtrls, Vcl.ComCtrls,System.Generics.Collections, Vcl.Imaging.pngimage;

type

TArrayForm = class(TForm)

StringGrid: TStringGrid;

Name: TLabel;

Array\_Type: TPanel;

Type\_Text: TLabel;

one\_dimensional: TPanel;

two\_dimensional: TPanel;

one\_dimensional\_set: TPanel;

UpDown1: TUpDown;

columns: TEdit;

Exit\_Button: TPanel;

Label\_Count: TLabel;

Label\_Data: TLabel;

ComboBox1: TComboBox;

Gen\_Button: TPanel;

Label2: TLabel;

UpDown2: TUpDown;

rows: TEdit;

Label1: TLabel;

TheoryBtn: TPanel;

Sort\_panel: TPanel;

Label3: TLabel;

ComboBox2: TComboBox;

Sort\_button: TPanel;

ComboBox3: TComboBox;

Find\_Panel: TPanel;

Label\_number: TLabel;

Edit\_Find: TEdit;

Label\_conclusion: TLabel;

Find\_Panel2: TPanel;

Memo\_find: TMemo;

ComboBox\_Find: TComboBox;

Find\_TypeLabel: TLabel;

Find\_Button: TPanel;

CodeBtn: TPanel;

Edit\_panel: TPanel;

Input\_ind\_Label: TLabel;

Input\_ind\_edit: TEdit;

element\_label: TLabel;

Label\_output: TLabel;

Edit\_Panel2: TPanel;

Show\_button: TPanel;

execute\_button: TPanel;

Move\_Label: TLabel;

ComboBox\_action: TComboBox;

IndexLabel: TLabel;

ValueLabel: TLabel;

IndexEdit: TEdit;

ValueEdit: TEdit;

Background: TImage;

ExitImg: TImage;

GenBtnImg: TImage;

CodebtnImg: TImage;

TheoryBtnImg: TImage;

GenBtnLabel: TLabel;

CodeBtnLabel: TLabel;

TheoryBtnlabel: TLabel;

Shape1: TShape;

Image2: TImage;

Label4: TLabel;

Image3: TImage;

Label5: TLabel;

Shape2: TShape;

Image4: TImage;

Label6: TLabel;

Shape3: TShape;

Shape4: TShape;

Image5: TImage;

Label7: TLabel;

Shape5: TShape;

Image6: TImage;

Label8: TLabel;

Shape6: TShape;

Image7: TImage;

Label9: TLabel;

Shape7: TShape;

ClearBtn: TPanel;

ClearBtnImg: TImage;

ClearBtnLabel: TLabel;

procedure Exit\_ButtonClick(Sender: TObject);

procedure Gen\_ButtonClick(Sender: TObject);

procedure UpDown1Click(Sender: TObject; Button: TUDBtnType);

procedure one\_dimensionalClick(Sender: TObject);

procedure two\_dimensionalClick(Sender: TObject);

procedure UpDown2Click(Sender: TObject; Button: TUDBtnType);

procedure StringGridSetEditText(Sender: TObject; ACol, ARow: Integer;

const Value: string);

procedure FormShow(Sender: TObject);

procedure SetMenuID(NewMenuID: Integer);

procedure Sort\_buttonClick(Sender: TObject);

procedure Sort();

procedure SortDescending();

procedure SelectionSort\_Increase();

procedure SelectionSort\_Decreasing();

procedure Bubblesort\_Increase();

procedure DisplayGrid();

procedure Bubblesort\_Decreasing();

procedure StringGridDrawCell(Sender: TObject; ACol, ARow: Integer;

Rect: TRect; State: TGridDrawState);

procedure FormHide(Sender: TObject);

procedure Find\_ButtonClick(Sender: TObject);

procedure Element\_Iteration();

procedure Binary\_search();

procedure ComboBox\_FindChange(Sender: TObject);

procedure Show\_buttonClick(Sender: TObject);

procedure execute\_buttonClick(Sender: TObject);

procedure TheoryBtnClick(Sender: TObject);

procedure CreateParams(var Params: TCreateParams) ; override;

procedure ClearBtnImgClick(Sender: TObject);

procedure CodebtnImgClick(Sender: TObject);

private

{ Private declarations }

procedure UpdateForm;

public

{ Public declarations }

end;

type

int=integer;

rel=real;

chr=char;

var

ArrayForm: TArrayForm;

Arr\_ID:integer;

Menu\_ID:integer;

index, CompareIndex1, CompareIndex2:integer;

exit\_activate:Boolean;

implementation

{$R \*.dfm}

uses SectionsFormUnit, TheoryFormUnit, CodeFormUnit;

procedure TArrayForm.Exit\_ButtonClick(Sender: TObject);

begin

SectionsForm.visible:=true;

ArrayForm.Visible:=false;

array\_type.Visible:=false;

Sort\_panel.Visible:=false;

combobox3.Visible:=false;

exit\_activate:=true;

end;

procedure TArrayForm.Find\_ButtonClick(Sender: TObject);

begin

memo\_find.lines.clear;

if Combobox\_find.ItemIndex=0 then

begin

Element\_Iteration();

end

else

begin

binary\_search();

end;

end;

procedure TArrayForm.FormHide(Sender: TObject);

var

i,j:integer;

begin

exit\_activate:=false;

for i := 0 to 11 do

for j := 0 to 3 do

stringgrid.Cells[i,j]:='';

stringgrid.RowCount:=1;

StringGrid.Height:=55;

label1.Visible:=false;

rows.Visible:=false;

updown2.Visible:=false;

Sort\_panel.Visible:=false;

combobox3.visible:=false;

Combobox\_Find.Visible:=false;

Find\_Panel.visible:=false;

Find\_Panel2.Visible:=false;

combobox1.Enabled:=true;

Edit\_panel.Visible:=false;

Edit\_panel2.Visible:=false;

valueedit.enabled:=true;

input\_ind\_edit.text:='';

Label\_output.Caption:='';

IndexEdit.Text:='';

ValueEdit.Text:='';

Memo\_find.Clear;

Edit\_find.Text:='';

end;

procedure TArrayForm.FormShow(Sender: TObject);

begin

exit\_activate:=false;

compareindex1:=-1;

compareindex2:=-1;

UpdateForm;

end;

procedure TArrayForm.SetMenuID(NewMenuID: Integer);

begin

Menu\_ID := NewMenuID;

UpdateForm;

end;

procedure TArrayForm.Show\_buttonClick(Sender: TObject);

var i:integer;

begin

if Input\_ind\_edit.Text = '' then //проверяю не пустое ли поле

begin

MessageDlg('Поле ввода не должно быть пустым!',mtError,[mbOk],0);

Exit;

end;

try

i := StrToInt(Input\_ind\_edit.Text);

except

on E: EConvertError do

begin

MessageDlg('Введено неверное значение. Пожалуйста, введите целое число.',mtError,[mbOk],0);

Exit;

end;

end;

i:=StrToInt(Input\_ind\_edit.text);

if (i<=stringgrid.colcount) and (i>0) then

begin

dec(i);

Label\_output.Caption:=stringgrid.Cells[i,0]

end

else

MessageDlg('Вы вышли за диапазон!',mtError,[mbOk],0);

end;

procedure TArrayForm.Element\_Iteration();

var complete: Boolean;

i:integer;

begin

complete:=false;

i:=0;

repeat

Compareindex1:=i;

displayGrid();

if stringgrid.Cells[i,0]=Edit\_Find.text then

complete:=true

else i:=i+1;

compareindex1:=-1;

DisplayGrid();

until (i>stringgrid.ColCount-1) or complete;

if complete then Memo\_find.Lines[0]:='Совпадение с элементом номер '+ IntToStr(i+1)

else Memo\_find.Lines[0]:='Совпадений нет';

end;

procedure TArrayForm.execute\_buttonClick(Sender: TObject);

var i,check:integer;

check1:Double;

check2:char;

Letters:Boolean;

begin

if combobox\_action.itemindex=0 then

begin

try

i:=StrToInt(IndexEdit.Text);

except

on E: EConvertError do

begin

MessageDlg('Поле индекса некорректно!',mtError,[mbOk],0);

Exit;

end;

end;

dec(i);

begin

case combobox1.itemindex of

0:begin

if not TryStrToInt(valueedit.Text,check) then

begin

MessageDlg('Введите целое число!',mtError,[mbOk],0);

exit;

end;

end;

1:begin

if not TryStrToFloat(valueedit.Text,check1) then

begin

MessageDlg('Введите вещественное число!',mtError,[mbOk],0);

exit;

end;

end;

2:begin

Letters:=True;

for i := 1 to length(valueedit.text) do

begin

if not CharInSet(Valueedit.text[i], ['A'..'Z', 'a'..'z']) then

begin

Letters := False;

Break;

end;

end;

if not Letters then

begin

MessageDlg('Пожалуйста, введите только буквы.',mtError,[mbOk],0);

exit;

end;

end;

end;

if (StrToInt(IndexEdit.Text)<0) or (StrToInt(IndexEdit.Text)>12) then

MessageDlg('Вы вышли за диапазон!',mtError,[mbOk],0)

else

stringgrid.Cells[i,0]:=valueedit.Text;

end;

end

else

begin

if stringgrid.width>=55 then

begin

updown1.Position:=updown1.position-1;

stringgrid.ColCount:=stringgrid.ColCount-1;

StringGrid.Width:=stringgrid.width-55;

stringgrid.Cells[stringgrid.colcount,0]:='';

end

else

MessageDlg('Нельзя удалить больше элементов!',mtError,[mbOk],0);

end;

end;

procedure TArrayForm.Binary\_search();

var top,bottom,operations,avg:integer;

complete:Boolean;

begin

if edit\_find.Text = '' then

begin

MessageDlg('Поле поиска не должно быть пустым!',mtError,[mbOk],0);

Exit;

end;

try

Sort();//отсортирую массив

except

on E: EConvertError do //проверка не пуст ли массив

begin

MessageDlg('Массив не заполен!',mtError,[mbOk],0);

Exit;

end;

end;

top:=0;

bottom:=stringgrid.ColCount-1;

complete:=false;

operations:=0;//задаем количество сравненией '0'

repeat

compareindex1:=top;

compareindex2:=bottom;

displaygrid();

if exit\_activate=true then //остановка цикла если вышли из формы

exit;

avg:=(bottom+top) div 2;

inc(operations);

if stringgrid.Cells[avg,0]=Edit\_Find.text then

complete:=true

else

begin

if StrToInt(Edit\_Find.text)<StrToInt(stringgrid.Cells[avg,0]) then

bottom:=avg-1

else

top:=avg+1;

end;

compareindex1:=-1;

compareindex2:=-1;

displaygrid();

until (top>bottom) or complete;

inc(avg);

if complete then Memo\_find.Lines[0]:='Совпадение с элементом номер '+ IntToStr(avg)+'. Выполнено '+IntToStr(operations)+' сравнений.'

else Memo\_find.Lines[0]:='Элемент не найден';

end;

procedure TArrayForm.Sort();

var n,i:integer;

List: TList<Integer>;

begin

List:=TList<Integer>.Create;

n:=updown1.position;

for i := 0 to n-1 do

List.Add(StrToInt(stringgrid.Cells[i,0]));

List.Sort;

for i:=0 to n-1 do

stringgrid.Cells[i,0]:=IntTostr(List[i]);

List.Free;

end;

procedure TArrayForm.SelectionSort\_Increase(); //метод прямого выбора по возрастанию

var i,j,min,ind:integer;

temp:string;

begin

for i:=0 to Stringgrid.Colcount-1 do//берем первый элемент

begin

min:=StrToInt(stringgrid.cells[i,0]);//задаем первый элемент как минимальный

ind:=i;//задаем первый индекс как минимальный

for j:=i+1 to Stringgrid.Colcount-1 do

begin

CompareIndex1 := i;

CompareIndex2 := j;

DisplayGrid();

if exit\_activate=true then //остановка цикла если вышли из формы

exit;

if StrToInt(Stringgrid.Cells[j,0])<min then//сравнение элементов с минимальным

begin

min:=StrToInt(Stringgrid.Cells[j,0]);

ind:=j;

end;

end;

temp:=stringgrid.Cells[i,0];

stringgrid.Cells[i,0]:=Stringgrid.Cells[ind,0];

stringgrid.Cells[ind,0]:=temp;//меняем местами элементы

//сброс подсветки

compareindex1:=-1;

compareindex2:=-1;

DisplayGrid();

end;

end;

procedure TArrayForm.SelectionSort\_Decreasing(); //метод прямого выбора по убыванию

var i,j,min,ind:integer;

temp:string;

begin

for i:=0 to Stringgrid.Colcount-1 do//берем первый элемент

begin

min:=StrToInt(stringgrid.cells[i,0]);//задаем первый элемент как минимальный

ind:=i;//задаем первый индекс как минимальный

for j:=i+1 to Stringgrid.Colcount-1 do

begin

CompareIndex1 := i;

CompareIndex2 := j;

DisplayGrid();

if exit\_activate=true then //остановка цикла если вышли из формы

exit;

if StrToInt(Stringgrid.Cells[j,0])>min then//сравнение элементов с минимальным

begin

min:=StrToInt(Stringgrid.Cells[j,0]);

ind:=j;

end;

end;

temp:=stringgrid.Cells[i,0];

stringgrid.Cells[i,0]:=Stringgrid.Cells[ind,0];

stringgrid.Cells[ind,0]:=temp;//меняем местами элементы

//сброс подсветки

compareindex1:=-1;

compareindex2:=-1;

DisplayGrid();

end;

end;

procedure TArrayForm.SortDescending();

var n,i:integer;

List: TList<Integer>;

begin

List:=TList<Integer>.Create;

n:=updown1.position;

for i := 0 to n-1 do

List.Add(StrToInt(stringgrid.Cells[i,0]));

List.Sort;

List.Reverse;

for i:=0 to n-1 do

stringgrid.Cells[i,0]:=IntTostr(List[i]);

List.Free;

end;

procedure TArrayForm.Bubblesort\_Increase();

var

i,j:integer;

temp: String;

Swapped: Boolean;

begin

for i := 0 to StringGrid.ColCount - 2 do

begin

Swapped := False;

for j := 0 to StringGrid.ColCount - 2 - i do

begin

compareindex1:=j;

compareindex2:=j+1;

// Отображение текущего состояния StringGrid

Displaygrid();

if StringGrid.Cells[j, 0] > StringGrid.Cells[j + 1, 0] then

begin

// Обмен значениями

temp := StringGrid.Cells[j, 0];

StringGrid.Cells[j, 0] := StringGrid.Cells[j + 1, 0];

StringGrid.Cells[j + 1, 0] := temp;

Swapped := True;

end;

//сброс подсветки

compareindex1:=-1;

compareindex2:=-1;

Displaygrid();

if exit\_activate=true then //остановка цикла если вышли из формы

exit

end;

//Если не было обменов, выходим из цикла

if not Swapped then

Break;

end;

end;

procedure TArrayForm.ClearBtnImgClick(Sender: TObject);

var i,j:integer;

begin

for i := 0 to 11 do

for j := 0 to 3 do

stringgrid.Cells[i,j]:='';

end;

procedure TArrayForm.CodebtnImgClick(Sender: TObject); //подгрузка примера кода

begin

case Menu\_ID of

1:begin //раздел "создание массивов

if stringgrid.RowCount=1 then //если одномерный массив

case combobox1.itemindex of

0:begin

CodeForm.SetCodeID(0); //целочисленный

CodeForm.Show;

end;

1:begin

CodeForm.SetCodeID(1); //вещественный

CodeForm.Show;

end;

2:begin

CodeForm.SetCodeID(2); //символьный

CodeForm.Show;

end;

end

else //если массив двумерный

begin

case combobox1.itemindex of

0:begin

CodeForm.SetCodeID(3);//целочисленный

CodeForm.Show;

end;

1:begin

CodeForm.SetCodeID(4);//вещественный

CodeForm.Show;

end;

2:begin

CodeForm.SetCodeID(5);//символьный

CodeForm.Show;

end;

end;

end;

end;

2:begin //раздел "сортировка массивов"

case combobox3.itemindex of

0:begin//процедурой

case combobox2.itemindex of

0:begin

CodeForm.SetCodeID(6);//по возрастанию

CodeForm.Show;

end;

1:begin

CodeForm.SetCodeID(7);//по убыванию

CodeForm.Show;

end;

end;

end;

1:begin//прямого выбора

case combobox2.itemindex of

0:begin

CodeForm.SetCodeID(8);//по возрастанию

CodeForm.Show;

end;

1:begin

CodeForm.SetCodeID(9);//по убыванию

CodeForm.Show;

end;

end;

end;

2:begin//метод пузырька

case combobox2.itemindex of

0:begin

CodeForm.SetCodeID(10);//по возрастанию

CodeForm.Show;

end;

1:begin

CodeForm.SetCodeID(11);//по убыванию

CodeForm.Show;

end;

end;

end;

end;

end;

3:begin //раздел "поиск в массиве"

case combobox\_find.itemindex of

0:begin //метод перебора элементов

case combobox1.ItemIndex of

0:begin

CodeForm.SetCodeID(12);//целочисленный

CodeForm.Show;

end;

1:begin

CodeForm.SetCodeID(13);//вещественный

CodeForm.Show;

end;

2:begin

CodeForm.SetCodeID(14);//символьный

CodeForm.Show;

end;

end;

end;

1:begin //бинарный поиск

CodeForm.SetCodeID(15);

CodeForm.Show;

end;

end;

end;

4:begin //раздел "Чтение и изменение элементов"

case combobox\_action.itemindex of //изменение элемента

0:begin

case combobox1.itemindex of

0:begin

CodeForm.SetCodeID(16); //целочисленный

CodeForm.Show;

end;

1:begin

CodeForm.SetCodeID(17); //вещественный

CodeForm.Show;

end;

2:begin

CodeForm.SetCodeID(18); //символьный

CodeForm.Show;

end;

end;

end;

1:begin //удаление элемента

CodeForm.SetCodeID(19);

CodeForm.Show;

end;

end;

end;

end;

end;

procedure TArrayForm.ComboBox\_FindChange(Sender: TObject);

begin

if combobox\_find.itemindex=1 then

begin

combobox1.ItemIndex:=0;

combobox1.Enabled:=false;

end

else

begin

combobox1.enabled:=true;

end;

end;

procedure TArrayForm.Bubblesort\_Decreasing();

var

i,j:integer;

temp: String;

Swapped: Boolean;

begin

for i := 0 to StringGrid.ColCount - 2 do

begin

Swapped := False;

for j := 0 to StringGrid.ColCount - 2 - i do

begin

compareindex1:=j;

compareindex2:=j+1;

// Отображение текущего состояния StringGrid

Displaygrid();

if StringGrid.Cells[j, 0] < StringGrid.Cells[j + 1, 0] then

begin

// Обмен значениями

temp := StringGrid.Cells[j, 0];

StringGrid.Cells[j, 0] := StringGrid.Cells[j + 1, 0];

StringGrid.Cells[j + 1, 0] := temp;

Swapped := True;

end;

//сброс подсветки

compareindex1:=-1;

compareindex2:=-1;

Displaygrid();

if exit\_activate=true then //остановка цикла если вышли из формы

exit

end;

// Если не было обменов, выходим из цикла

if not Swapped then

Break;

end;

end;

procedure TArrayForm.DisplayGrid();

begin

StringGrid.invalidate; //Обновление содержимого stringgrid

Application.ProcessMessages;//обработка всех сообщений в очереди

sleep(500);//задержка 500 мс между проходами элементов

end;

procedure TArrayForm.UpdateForm;

begin

if Menu\_ID=1 then //если нажата кнопка то подсвечиваются соответсвующие элементы

begin

Name.Caption:='Создание массива';

Label\_Data.Caption:='Выбор типа данных массива:';

array\_type.Visible:=true;

ArrayForm.Caption:='Создание массива';

end

else

if Menu\_ID=2 then

begin

Name.Caption:='Сортировка массива';

Label\_Data.Caption:='Тип сортировки массива:';

ArrayForm.Caption:='Сортировка массива';

combobox3.visible:=true;

Sort\_panel.Visible:=true;

Arr\_ID:=0;

combobox1.ItemIndex:=0;

end

else

if Menu\_ID=3 then

begin

Name.Caption:='Поиск в массиве';

Label\_Data.Caption:='Выбор типа данных массива:';

ArrayForm.Caption:='Поиск в массиве';

Find\_panel.Visible:=True;

Find\_panel2.Visible:=true;

Combobox\_Find.Visible:=True;

Arr\_ID:=0;

index:=0;

end

else

if Menu\_ID=4 then

begin

Name.Caption:='Чтение и изменение массива';

Label\_Data.Caption:='Выбор типа данных массива:';

ArrayForm.Caption:='Чтение и изменение элементов';

Edit\_panel.Visible:=true;

Edit\_panel2.Visible:=true;

end;

end;

procedure TArrayForm.Gen\_ButtonClick(Sender: TObject);

var n,m,i,j:integer;

a:array of integer;

b:array of real;

c:array of char;

a1:array of array of integer;

b1:array of array of real;

c1:array of array of char;

begin

randomize;

n:=updown1.Position;

m:=updown2.Position;

index:=combobox1.ItemIndex;

if Arr\_ID=0 then

begin

//одномерный массив

setlength(a,n);

setlength(b,n);

setlength(c,n);

if index=0 then//заполнение целыми числами

begin

for i:=0 to n-1 do

begin

a[i]:=random(100);

stringgrid.cells[i,0]:=IntToStr(a[i]);

end;

end

else

if index=1 then //заполнение вещественными числами

begin

for i:=0 to n-1 do

begin

b[i]:=random;

stringgrid.cells[i,0]:=FloatToStrF(b[i],ffFixed, 8, 2);

end;

end

else

if index=2 then //заполнение символами

begin

for i:=0 to n-1 do

begin

c[i]:=chr(random(26)+65);

stringgrid.cells[i,0]:=c[i];

end;

end;

end

else

begin //двумерный массив

setlength(a1,n,m);

setlength(b1,n,m);

setlength(c1,n,m);

if index=0 then //заполнение целыми числами

begin

for i:=0 to n-1 do

for j:=0 to m-1 do

begin

a1[i,j]:=random(100);

stringgrid.cells[i,j]:=IntToStr(a1[i,j]);

end;

end

else

if index=1 then //заполнение вещественными числами

begin

for i:=0 to n-1 do

for j := 0 to m-1 do

begin

b1[i,j]:=random;

stringgrid.cells[i,j]:=FloatToStrF(b1[i,j],ffFixed, 8, 2);

end;

end

else

if index=2 then //заполнение символами

begin

for i:=0 to n-1 do

for j := 0 to m-1 do

begin

c1[i,j]:=chr(random(26)+65);

stringgrid.cells[i,j]:=c1[i,j];

end;

end;

end;

end;

procedure TArrayForm.one\_dimensionalClick(Sender: TObject);

begin

label1.Visible:=false;

rows.Visible:=false;

updown2.Visible:=false;

StringGrid.Height:=55;

stringgrid.RowCount:=1;

Arr\_ID:=0;

end;

procedure TArrayForm.TheoryBtnClick(Sender: TObject);

begin

TheoryForm.show;

end;

procedure TArrayForm.Sort\_buttonClick(Sender: TObject);

var n,i:integer;

List: TList<Integer>;

begin

if combobox3.ItemIndex=0 then

begin

if combobox2.ItemIndex=0 then

begin

try

Sort();

except //проверка заполнен ли массив

on E: EConvertError do

begin

MessageDlg('Массив не заполнен!',mtError,[mbOk],0);

Exit;

end;

end;

end

else

begin

try

SortDescending();

except //проверка заполнен ли массив

on E: EConvertError do

begin

MessageDlg('Массив не заполнен!',mtError,[mbOk],0);

Exit;

end;

end;

end;

end

else

if combobox3.ItemIndex=1 then

begin

if combobox2.ItemIndex=0 then

begin

try

SelectionSort\_Increase();

except

on E: EConvertError do

begin

MessageDlg('Массив не заполнен!',mtError,[mbOk],0);

Exit;

end;

end;

end

else

begin

try

SelectionSort\_Decreasing();

except

on E: EconvertError do

begin

MessageDlg('Массив не заполнен!',mtError,[mbOk],0);

Exit;

end;

end;

end;

end

else

if combobox3.ItemIndex=2 then

begin

if combobox2.ItemIndex=0 then

begin

Bubblesort\_Increase();

end

else

begin

Bubblesort\_Decreasing();

end;

end;

end;

procedure TArrayForm.StringGridDrawCell(Sender: TObject; ACol, ARow: Integer;

Rect: TRect; State: TGridDrawState);

begin

if (ACol=Compareindex1) or (ACol=Compareindex2) then

begin

stringgrid.Canvas.brush.Color:=clYellow;

stringgrid.Canvas.FillRect(rect);

end;

StringGrid.Canvas.TextRect(Rect, Rect.Left + 6, Rect.Top + 18, StringGrid.Cells[ACol, ARow]);

end;

procedure TArrayForm.StringGridSetEditText(Sender: TObject; ACol, ARow: Integer;

const Value: string);

var num,i:integer;

num1:Double;

num2:char;

Letters:boolean;

begin //проверка вводимого типа данных

if combobox1.ItemIndex=0 then

begin

if not TryStrToInt(value,num) then

begin

MessageDlg('Пожалуйста, введите целое число.',mtError,[mbOk],0);

StringGrid.Cells[ACol, ARow] := ''; //очистка ячейки, если введено не число

end;

end

else

if combobox1.ItemIndex=1 then

begin

if not TryStrToFloat(value,num1) then

begin

MessageDlg('Пожалуйста, введите вещественное число.',mtError,[mbOk],0);

StringGrid.Cells[ACol, ARow] := ''; //очистка ячейки, если введено не вещественное число

end;

end

else

if combobox1.ItemIndex=2 then

begin

Letters:=True;

for i := 1 to length(value) do

begin

if not CharInSet(Value[i], ['A'..'Z', 'a'..'z']) then

begin

Letters := False;

Break;

end;

end;

if not Letters then

begin

MessageDlg('Пожалуйста, введите только буквы.',mtError,[mbOk],0);

StringGrid.Cells[ACol, ARow] := ''; //очистка ячейки, если введено не буквенное значение

end;

end;

end;

procedure TArrayForm.two\_dimensionalClick(Sender: TObject);

begin

label1.Visible:=true;

rows.Visible:=true;

updown2.Visible:=true;

Arr\_ID:=1;

stringgrid.RowCount:=2;

stringgrid.Height:=110;

end;

procedure TArrayForm.UpDown1Click(Sender: TObject; Button: TUDBtnType);

begin

StringGrid.ColCount:=updown1.position;

case stringgrid.colcount of //изменяю длину поля

2:StringGrid.Width:=110;

3:StringGrid.Width:=165;

4:StringGrid.Width:=220;

5:StringGrid.Width:=275;

6:StringGrid.Width:=330;

7:StringGrid.Width:=385;

8:StringGrid.Width:=440;

9:StringGrid.Width:=495;

10:StringGrid.Width:=550;

11:StringGrid.Width:=605;

12:StringGrid.Width:=660;

end;

end;

procedure TArrayForm.UpDown2Click(Sender: TObject; Button: TUDBtnType);

begin

StringGrid.RowCount:=updown2.position;

case StringGrid.RowCount of //изменяю высоту поля

1:StringGrid.Height:=55;

2:StringGrid.Height:=110;

3:StringGrid.Height:=165;

4:StringGrid.Height:=220;

end;

end;

procedure TArrayForm.CreateParams(var Params: TCreateParams) ;// процедура для того чтобы ярлык в панели задач не пропадал

begin

inherited;

Params.ExStyle := Params.ExStyle or WS\_EX\_APPWINDOW;

Params.WndParent := 0;

end;

end.

unit CodeFormUnit;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.OleCtrls, SHDocVw, Vcl.StdCtrls;

type

TCodeForm = class(TForm)

WebBrowser1: TWebBrowser;

procedure SetCodeID(NewCodeID: Integer);

procedure FormShow(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

CodeForm: TCodeForm;

Code\_ID: integer;

implementation

{$R \*.dfm}

uses ArrayFormUnit;

procedure TCodeForm.FormShow(Sender: TObject);

var

filepath:string;

begin

filepath:=extractfilepath(application.ExeName);

case Code\_ID of

0:webbrowser1.navigate(PChar(filepath + 'codes\Создание массивов\одномерные\целочисленный.htm'));

1:webbrowser1.navigate(PChar(filepath + 'codes\Создание массивов\одномерные\вещественный.htm'));

2:webbrowser1.navigate(PChar(filepath + 'codes\Создание массивов\одномерные\символьный.htm'));

3:webbrowser1.navigate(PChar(filepath + 'codes\Создание массивов\двумерные\целочисленный.htm'));

4:webbrowser1.navigate(PChar(filepath + 'codes\Создание массивов\двумерные\вещественный.htm'));

5:webbrowser1.navigate(PChar(filepath + 'codes\Создание массивов\двумерные\символьный.htm'));

6:webbrowser1.navigate(PChar(filepath + 'codes\Сортировка массива\по возрастанию\процедурой.htm'));

7:webbrowser1.navigate(PChar(filepath + 'codes\Сортировка массива\по убыванию\процедурой.htm'));

8:webbrowser1.navigate(PChar(filepath + 'codes\Сортировка массива\по возрастанию\прямого выбора.htm'));

9:webbrowser1.navigate(PChar(filepath + 'codes\Сортировка массива\по убыванию\прямого выбора.htm'));

10:webbrowser1.navigate(PChar(filepath + 'codes\Сортировка массива\по возрастанию\пузырьком.htm'));

11:webbrowser1.navigate(PChar(filepath + 'codes\Сортировка массива\по убыванию\пузырьком.htm'));

12:webbrowser1.navigate(PChar(filepath + 'codes\Поиск в массиве\Перебор элементов\целочисленный.htm'));

13:webbrowser1.navigate(PChar(filepath + 'codes\Поиск в массиве\Перебор элементов\вещественный.htm'));

14:webbrowser1.navigate(PChar(filepath + 'codes\Поиск в массиве\Перебор элементов\символьный.htm'));

15:webbrowser1.navigate(PChar(filepath + 'codes\Поиск в массиве\бинарный поиск.htm'));

16:webbrowser1.navigate(PChar(filepath + 'codes\Чтение и изменение массива\изменение элемента\целочисленный.htm'));

17:webbrowser1.navigate(PChar(filepath + 'codes\Чтение и изменение массива\изменение элемента\вещественный.htm'));

18:webbrowser1.navigate(PChar(filepath + 'codes\Чтение и изменение массива\изменение элемента\символьный.htm'));

19:webbrowser1.navigate(PChar(filepath + 'codes\Чтение и изменение массива\удаление элемента.htm'));

end;

end;

procedure TCodeForm.SetCodeID(NewCodeID: Integer);

begin

Code\_ID := NewCodeID;

end;

end.