МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра ЭВМ

Отчёт

### Лабораторная работа № 8 по дисциплине

«Программирование»

«Знакомство со средой программирования Lazarus»

Выполнил студент группы ИВТб-1301 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К. А.

## Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Чистяков Г.А.

Киров 2021

**Цель:** познакомиться с парадигмой событийно-ориентированного программирования, получить базовые навыки создания программ с графическим интерфейсом пользователя.

**Задание:**

1. Разработать программу «Текстовый редактор».
2. Реализовать возможность работы с начертанием шрифта, вывода текста на печать, подсчета статистики,
3. В обязательном порядке должны быть реализованы функции «открыть», «создать», «сохранить» и «сохранить как».
4. Реализация дополнительных возможностей будет оцениваться в индивидуальном порядке.

**Листинг:**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, Menus,

ActnList, ComCtrls, Clipbrd, PrintersDlgs;

type

{ TForm1 }

TForm1 = class(TForm)

InfAction: TAction;

MenuItem18: TMenuItem;

MenuItem19: TMenuItem;

MenuItem20: TMenuItem;

PrintAction: TAction;

ApplicationProperties1: TApplicationProperties;

FontAction: TAction;

FontDialog1: TFontDialog;

MenuItem16: TMenuItem;

MenuItem17: TMenuItem;

PrintDialog1: TPrintDialog;

PrinterSetupDialog1: TPrinterSetupDialog;

WordWrapAction: TAction;

FindAction: TAction;

FindDialog1: TFindDialog;

MenuItem15: TMenuItem;

SelectAllAction: TAction;

DeleteAction: TAction;

MenuItem13: TMenuItem;

MenuItem14: TMenuItem;

PasteAction: TAction;

CopyAction: TAction;

CutAction: TAction;

MenuItem10: TMenuItem;

MenuItem11: TMenuItem;

MenuItem12: TMenuItem;

UndoAction: TAction;

NewAction: TAction;

SaveAction: TAction;

SaveAsAction: TAction;

MenuItem4: TMenuItem;

MenuItem5: TMenuItem;

MenuItem6: TMenuItem;

MenuItem7: TMenuItem;

MenuItem8: TMenuItem;

MenuItem9: TMenuItem;

OpenAction: TAction;

ExitAction: TAction;

ActionList1: TActionList;

MainMenu1: TMainMenu;

Memo1: TMemo;

MenuItem1: TMenuItem;

MenuItem2: TMenuItem;

MenuItem3: TMenuItem;

OpenDialog1: TOpenDialog;

SaveDialog1: TSaveDialog;

StatusBar1: TStatusBar;

procedure ApplicationProperties1Hint(Sender: TObject);

procedure ApplicationProperties1Idle(Sender: TObject; var Done: Boolean);

procedure CopyActionExecute(Sender: TObject);

procedure CutActionExecute(Sender: TObject);

procedure CutActionUpdate(Sender: TObject);

procedure DeleteActionExecute(Sender: TObject);

procedure DeleteActionUpdate(Sender: TObject);

procedure FontActionExecute(Sender: TObject);

procedure FormCloseQuery(Sender: TObject; var CanClose: boolean);

procedure FormCreate(Sender: TObject);

procedure InfActionExecute(Sender: TObject);

procedure NewActionExecute(Sender: TObject);

procedure PasteActionExecute(Sender: TObject);

procedure PasteActionUpdate(Sender: TObject);

procedure PrintActionExecute(Sender: TObject);

procedure SaveActionExecute(Sender: TObject);

procedure SaveAsActionExecute(Sender: TObject);

procedure ExitActionExecute(Sender: TObject);

procedure OpenActionExecute(Sender: TObject);

procedure SelectAllActionExecute(Sender: TObject);

procedure UndoActionExecute(Sender: TObject);

procedure UndoActionUpdate(Sender: TObject);

procedure WordWrapActionExecute(Sender: TObject);

private

FLastSearch: integer;

FOpenFile: string;

function CloseCurrentDocument: Boolean;

public

end;

var

Form1: TForm1;

implementation

{$R \*.lfm}

// проверка возможности закрытия программы

function TForm1.CloseCurrentDocument: Boolean;

begin

CloseCurrentDocument:= True;

if Memo1.Modified then

begin

case messagedlg('Сохранить изменения?', mtConfirmation, mbYesNoCancel,0) of

mrYes: SaveAction.Execute;

mrCancel: CloseCurrentDocument:= False;

end;

end;

end;

{ TForm1 }

// закрытие программы

procedure TForm1.ExitActionExecute(Sender: TObject);

begin

Close;

end;

// функция «сохранить как»

procedure TForm1.SaveAsActionExecute(Sender: TObject);

begin

If SaveDialog1.Execute then

begin

FOpenFile:= SaveDialog1.FileName;

Memo1.Lines.savetofile(FOpenFile);

Memo1.Modified:= False;

end;

end;

// функция «сохранить»

procedure TForm1.SaveActionExecute(Sender: TObject);

begin

If FOpenFile <> '' then

begin

Memo1.Lines.SaveToFile(FOpenFile);

Memo1.Modified:= False;

end

else

SaveAsAction.Execute;

Form1.Caption:= FOpenFile;

Memo1.Modified:= False;

end;

procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: boolean);

begin

CanClose:= CloseCurrentDocument;

end;

// объявление заголовка

procedure TForm1.FormCreate(Sender: TObject);

begin

Form1.Caption:= 'Безымянный - Блокнот';

end;

// О программе

procedure TForm1.InfActionExecute(Sender: TObject);

begin

MessageDlg('Разработал студент ИВТ-11 Жеребцов К. А. 2021', mtInformation, [mbOk], 0);

end;

// функция «копировать»

procedure TForm1.CopyActionExecute(Sender: TObject);

begin

Memo1.CopyToClipboard;

end;

// отслеживание статистики

procedure TForm1.ApplicationProperties1Idle(Sender: TObject; var Done: Boolean);

Const

modified: array[Boolean] of string = ('', 'Modified');

begin

StatusBar1.Panels[0].Text:=Format('Стр %d, стлб %d',

[succ(Memo1.CaretPos.Y),succ(Memo1.CaretPos.X)]);

StatusBar1.Panels[1].Text:= modified[Memo1.Modified];

end;

// вывод подсказок

procedure TForm1.ApplicationProperties1Hint(Sender: TObject);

begin

StatusBar1.Panels[2].Text:= Application.Hint;

end;

// функция «вырезать»

procedure TForm1.CutActionExecute(Sender: TObject);

begin

Memo1.CutToClipboard;

end;

procedure TForm1.CutActionUpdate(Sender: TObject);

begin

TAction(Sender).Enabled:= Memo1.SelLength > 0;

end;

//функция удалить

procedure TForm1.DeleteActionExecute(Sender: TObject);

begin

Memo1.ClearSelection;

end;

// проверка на выделение текста

procedure TForm1.DeleteActionUpdate(Sender: TObject);

begin

TAction(Sender).Enabled:= Memo1.SelLength > 0;

end;

// работа со шрифтом

procedure TForm1.FontActionExecute(Sender: TObject);

begin

FontDialog1.Font.Assign(Memo1.Font);

if FontDialog1.Execute then

Memo1.Font.Assign(FontDialog1.Font);

end;

// перенос по словам

procedure TForm1.WordWrapActionExecute(Sender: TObject);

const

scrolls: array[Boolean] of TScrollStyle = (ssBoth, ssVertical);

begin

Memo1.WordWrap:= WordWrapAction.Checked;

Memo1.ScrollBars:= scrolls[Memo1.WordWrap];

end;

// функция «создать»

procedure TForm1.NewActionExecute(Sender: TObject);

begin

If CloseCurrentDocument then

begin

Memo1.Lines.clear;

FOpenFile:= '';

Form1.Caption:= 'Безымянный - Блокнот';

Memo1.Modified:= False;

end;

end;

// функция «вставить»

procedure TForm1.PasteActionExecute(Sender: TObject);

begin

Memo1.PasteFromClipboard;

end;

// проверка на выделение текста

procedure TForm1.PasteActionUpdate(Sender: TObject);

begin

PasteAction.Enabled:= Clipboard.HasFormat(CF\_TEXT);

end;

// печать

procedure TForm1.PrintActionExecute(Sender: TObject);

begin

If CloseCurrentDocument then

begin

If PrinterSetupDialog1.Execute then

PrintDialog1.Execute;

end;

end;

// функция «открыть»

procedure TForm1.OpenActionExecute(Sender: TObject);

begin

If CloseCurrentDocument and OpenDialog1.Execute then

begin

FOpenFile:= OpenDialog1.FileName;

Memo1.Lines.LoadFromFile(FOpenFile);

Form1.Caption:= FOpenFile;

Memo1.Modified:= False;

end;

end;

// функция «выделить все»

procedure TForm1.SelectAllActionExecute(Sender: TObject);

begin

Memo1.SelectAll;

end;

// функция «отменить последнее действие»

procedure TForm1.UndoActionExecute(Sender: TObject);

begin

Memo1.Undo;

end;

// проверка на возможность отмены действия

procedure TForm1.UndoActionUpdate(Sender: TObject);

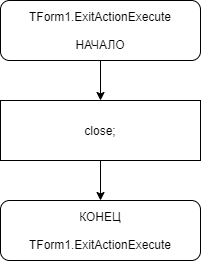
begin

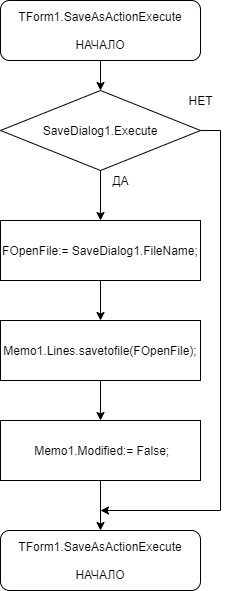
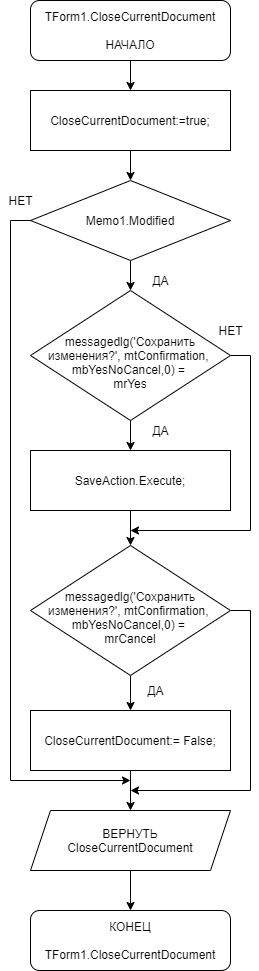
UndoAction.Enabled:= memo1.CanUndo;

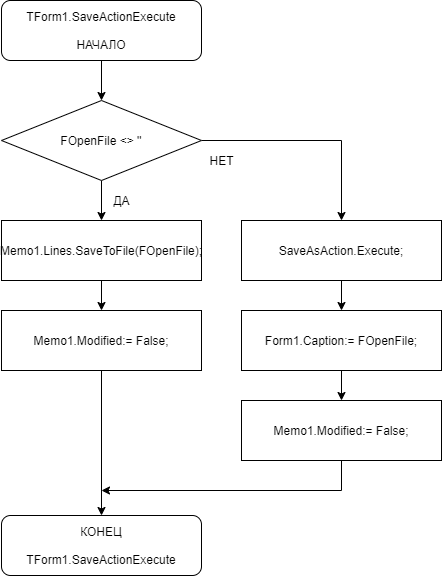
end;

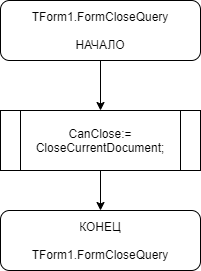
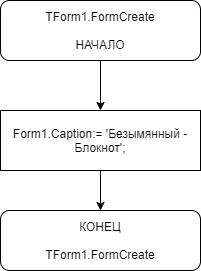
end.

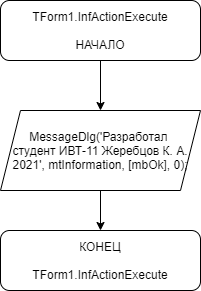
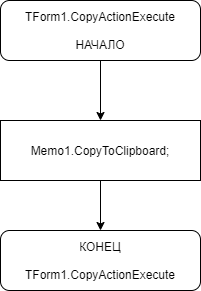
**Схема алгоритма:**

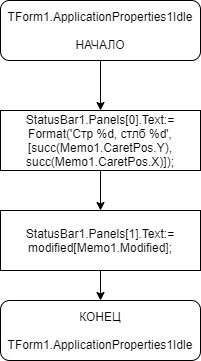
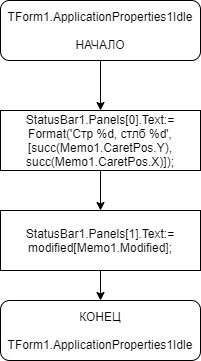


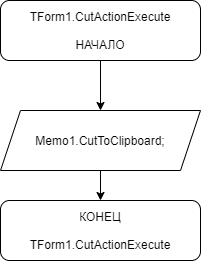
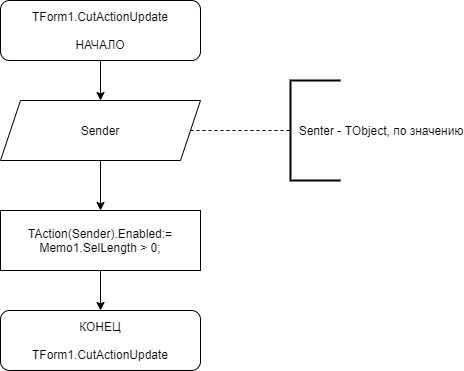
 

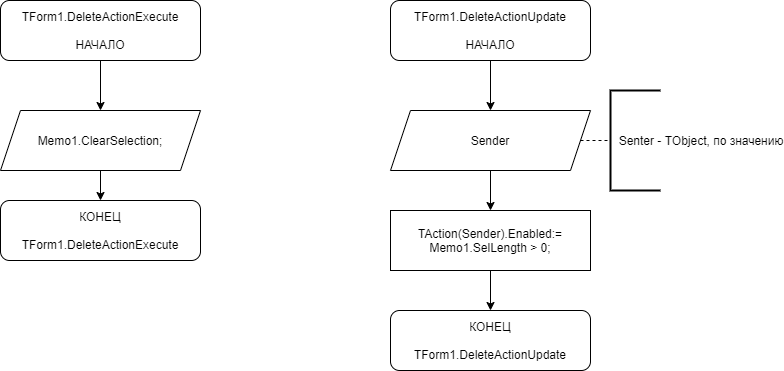
****

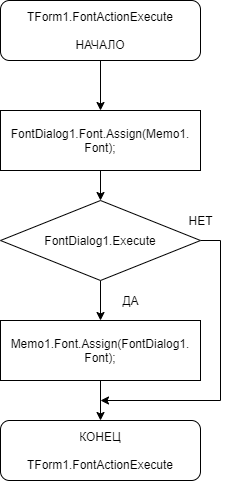
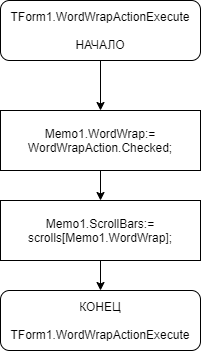
****

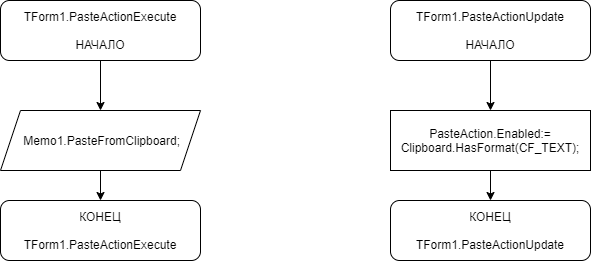
****

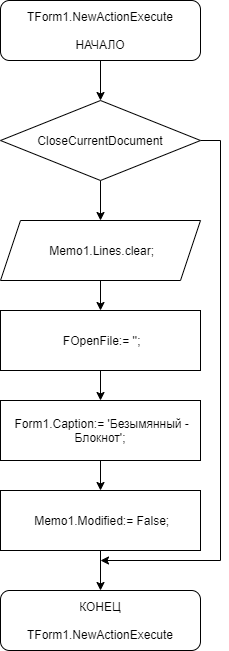
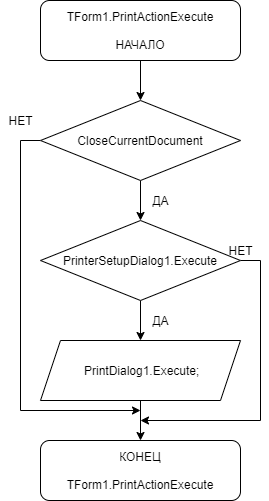
****

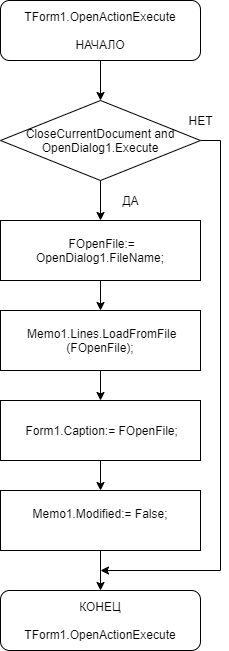
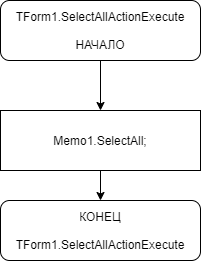
** **

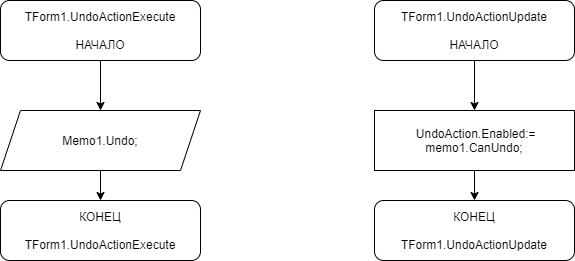
****

** **

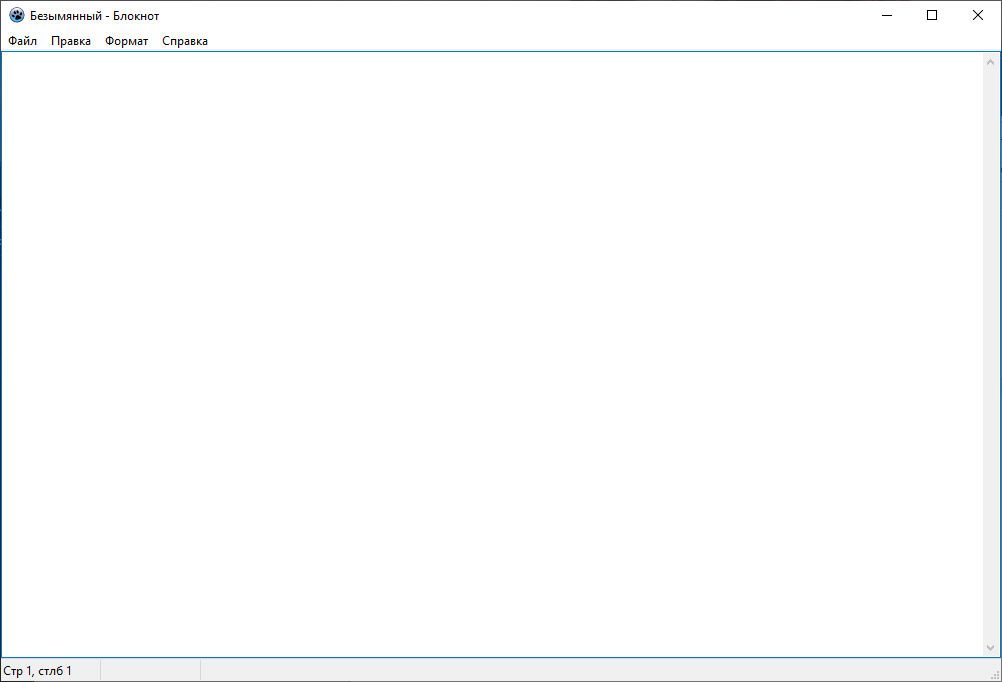
****

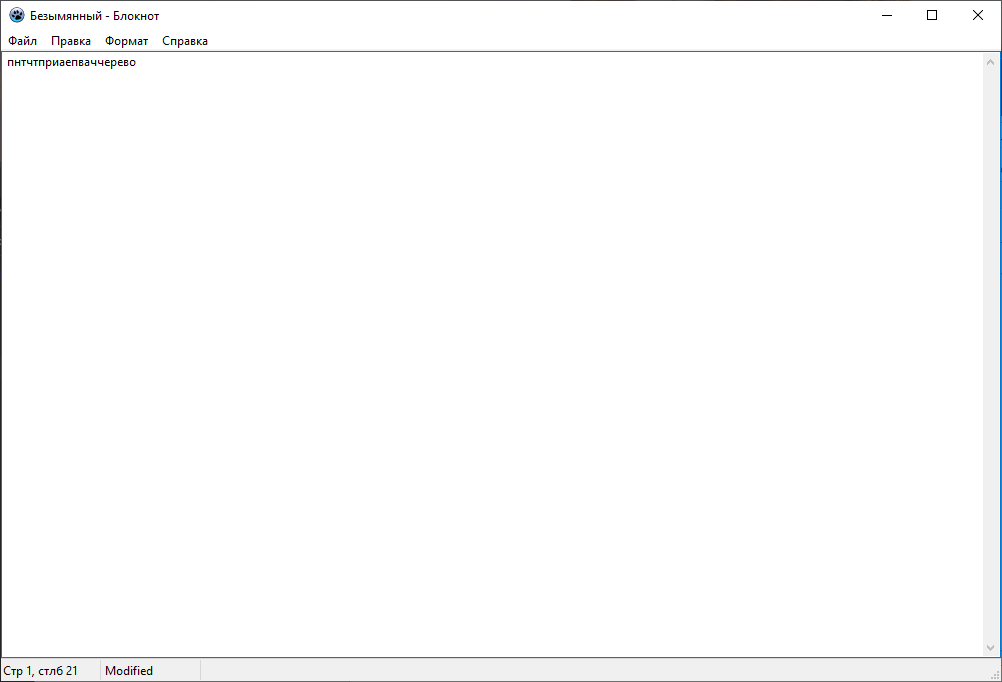
****

****

****

**Экранные формы:**

****

****

**Вывод:**

Были получены базовые навыки событийно-ориентированного программирования. Суть заключается в том, чтобы обработать все возможные действия пользователя путем написания для каждого подпрограммы. Также были получены базовые навыки создания программ с графическим интерфейсом пользователя. Он создается с помощью компонентов среды программирования Lazarus, такие как «TMainMenu», «TStatusBar», «TButton» и так далее.