Балтийский государственный технический университет «Военмех» им. Д. Ф. Устинова

Кафедра И5

«Информационные системы и программная инженерия»

**ЛАБОРАТОРНАЯ РАБОТА № 2**

По дисциплине **«Визуальное программирование»**

На тему

**«РАБОТА С КОМПОНЕНТАМИ В РЕЖИМАХ**

**DESIGN-TIME И RUN-TIME»**

***Вариант № 4***

Выполнил:

Студент Васильев Н. А.

Группа И967

**Преподаватель**:

Ракова И. К.

Санкт-Петербург

2018

**Цель работы**:

Научиться создавать приложения в режиме design-time, а также приложения с настраиваемым интерфейсом пользователя с применением режима run-time.

**Задание:**

Необходимо разработать приложение, позволяющее производить загрузку стандартных приложений Windows. Отображение списка приложений может производиться двумя способами:

1) в списке выбора (компонент ListBox);

2) с помощью кнопок (компоненты Button).

В ini-файле хранятся следующие данные: тип отображения (1 или 2), список заголовков приложений и список командных строк для запуска приложений.

**Текст программы:**

*MainFormUnit:*

unit MainFormUnit;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls, DesignTimeUnit, RunTimeUnit;

type

{ TMainForm }

TMainForm = class(TForm)

Design\_button: TButton;

Runtime\_button: TButton;

procedure designClick(Sender: TObject);

procedure runtimeClick(Sender: TObject);

private

public

end;

var

MainForm: TMainForm;

implementation

{$R \*.lfm}

{ TMainForm }

procedure TMainForm.designClick(Sender: TObject);

begin

DesignTimeForm.Show;

end;

procedure TMainForm.runtimeClick(Sender: TObject);

begin

RunTimeForm.Show;

end;

end.

*RunTimeUnit:*

unit RunTimeUnit;

{$mode objfpc}{$H+}

interface

uses Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, ComponentManagerUnit;

type

{ TRunTimeForm }

TRunTimeForm = class(TForm)

procedure ShowForm(Sender: TObject);

procedure CloseForm(Sender: TObject);

private

ComponentManager: TComponentManager;

public

end;

var

RunTimeForm: TRunTimeForm;

implementation

{$R \*.lfm}

procedure TRunTimeForm.ShowForm(Sender: TObject);

begin

ComponentManager := TComponentManager.Create;

end;

procedure TRunTimeForm.CloseForm(Sender: TObject);

begin

ComponentManager.Destroy;

end;

end.

*DesignTimeUnit:*

unit DesignTimeUnit;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls, ShellApi;

type

{ TDesignTimeForm }

TDesignTimeForm = class(TForm)

Button1: TButton;

Button2: TButton;

Button3: TButton;

ListBox1: TListBox;

procedure ButtonClick(Sender: TObject);

procedure ListClick(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: integer);

procedure SpacePress(Sender: TObject; var Key: Word; Shift: TShiftState);

private

public

end;

var

DesignTimeForm: TDesignTimeForm;

implementation

{$R \*.lfm}

{ TDesignTimeForm }

procedure TDesignTimeForm.ButtonClick(Sender: TObject);

begin

if Sender = Button1 then

ShellExecute(0, 'open', 'mspaint', nil, nil, 1)

else if Sender = Button2 then

ShellExecute(0, 'open', 'notepad', nil, nil, 1)

else if Sender = Button3 then

ShellExecute(0, 'open', 'msconfig', nil, nil, 1);

end;

procedure TDesignTimeForm.ListClick(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: integer);

var

Point: TPoint;

Index: Integer;

begin

if (Button <> mbleft) or not(ssDouble in Shift) then

Exit;

Point.X := X;

Point.Y := Y;

Index:= ListBox1.ItemAtPos(Point, True);

Case Index of

0: ShellExecute(0, 'open', 'mspaint', nil, nil, 1);

1: ShellExecute(0, 'open', 'notepad', nil, nil, 1);

2: ShellExecute(0, 'open', 'msconfig', nil, nil, 1);

end;

end;

procedure TDesignTimeForm.SpacePress(Sender: TObject; var Key: Word;

Shift: TShiftState);

begin

if (Key = 32) or (Key = 13) then begin

Case ListBox1.ItemIndex of

0: ShellExecute(0, 'open', 'mspaint', nil, nil, 1);

1: ShellExecute(0, 'open', 'notepad', nil, nil, 1);

2: ShellExecute(0, 'open', 'msconfig', nil, nil, 1);

end;

end;

end;

end.

*ComponentManagerUnit:*

unit ComponentManagerUnit;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, StdCtrls, FileUtil, Forms, Controls, Graphics, Dialogs, IniFiles, ShellApi;

type

TComponentManager = class

private

buttonsList: array of TButton;

myListbox: TListBox;

Ini: Tinifile;

NAMES, CMD\_LINES: array of string;

size: integer;

procedure useButtons();

procedure useListbox();

procedure ListClick(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: integer);

procedure SpacePress(Sender: TObject; var Key: Word; Shift: TShiftState);

procedure ButtonClick(Sender: TObject);

public

constructor Create();

destructor Destroy(); override;

end;

implementation

uses RunTimeUnit;

//Клик по кнопке

procedure TComponentManager.ButtonClick(Sender: TObject);

begin

ShellExecute(0, 'open', PChar(CMD\_LINES[TButton(Sender).Tag]), nil, nil, 1);

end;

//Пробел или Enter по списку

procedure TComponentManager.SpacePress(Sender: TObject; var Key: Word; Shift: TShiftState);

begin

if (Key = 32) or (Key = 13) then

ShellExecute(0, 'open', PChar(CMD\_LINES[myListbox.ItemIndex]), nil, nil, 1);

end;

//Клик по списку

procedure TComponentManager.ListClick(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: integer);

var

Point: TPoint;

Index: Integer;

begin

if (Button <> mbleft) or not(ssDouble in Shift) then

Exit;

Point.X := X;

Point.Y := Y;

Index := myListbox.ItemAtPos(Point, True);

if index > -1 then

ShellExecute(0, 'open', PChar(CMD\_LINES[Index]), nil, nil, 1);

end;

//Используем кнопки

procedure TComponentManager.useButtons();

var i: integer;

begin

setlength(buttonsList, size);

for i:= 0 to size-1 do begin

buttonsList[i] := TButton.Create(RunTimeForm);

buttonsList[i].Parent := RunTimeForm;

buttonsList[i].Width := 304;

buttonsList[i].Height := 30;

buttonsList[i].Left := 8;

buttonsList[i].Tag := i;

buttonsList[i].Top := 8 + i \* 30;

buttonsList[i].Caption := NAMES[i];

buttonsList[i].OnCLick := @ButtonClick;

end;

end;

//Используем список

procedure TComponentManager.useListbox();

var i: integer;

begin

myListbox := TListBox.Create(RunTimeForm);

myListbox.Parent := RunTimeForm;

myListbox.Width := 304;

myListbox.Top := 8;

myListbox.Left := 8;

myListbox.Height := 200;

for i:=0 to size-1 do

myListBox.Items.Add(NAMES[i]);

myListbox.OnMouseUp := @ListClick;

myListbox.OnKeyUp := @SpacePress;

end;

//конструктор

constructor TComponentManager.Create;

var OUTPUT\_TYPE: byte;

begin

Ini:= TiniFile.Create(extractfilepath(Application.ExeName)+'myinifile.ini');

OUTPUT\_TYPE := Ini.ReadInteger('OUTPUT', 'OUTPUT\_TYPE', 1);

size := 0;

repeat

setlength(NAMES, size + 1);

setlength(CMD\_LINES, size + 1);

NAMES[size] := Ini.ReadString('NAMES', 'NAME\_'+IntToStr(size), 'null');

CMD\_LINES[size] := Ini.ReadString('CMD\_LINES', 'LINE\_'+IntToStr(size), 'null');

size := size + 1;

until (NAMES[size-1] = 'null') or (CMD\_LINES[size-1] = 'null');

size := size - 1;

setlength(NAMES, size);

setlength(CMD\_LINES, size);

if OUTPUT\_TYPE = 2 then

useButtons()

else

useListbox();

end;

//деструктор

destructor TComponentManager.Destroy;

var i: integer;

begin

for i := 0 to size-1 do

buttonsList[i].Free;

myListbox.Free;

Ini.Free;

end;

end.

**Результаты работы программы:**



