Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №8**

**«Основы событийно-ориентированного программирования»**

**ПО «МДК 05.02. Разработка кода**

**информационных систем»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Четвериков Д.А.

Преподаватель:

Сергеева Е.Г.

Киров

2025

**Цель:** получение базовых навыков реализации приложений с графическим интерфейсом пользователя на основе событийно-ориентированного программирования.

**Задание (Вариант 8):**

1. Реализовать приложение «Калькулятор площади фигур».
2. Предусмотреть выбор из фиксированного списка фигур(не менее 15).

**Описание алгоритма:**

Unit1

1. TForm1.FormCreate иницилизация списка фигур
2. Заполнение ComboBox
3. Основаня функция для обработки нажатия на кнопку
4. Она считает площадь по введённым параметрам, в зависимости от того, какая фигура была выбрана.

**Описание элементов**

TLabel - используется для подписей полей, в которые записываются значения.

Tbutton - используется для создания кнопок для рассчета итоговой суммы, очистки полей, вывода окна с информацией.

TEdit - используется для ввода значений.

TCombobox - используется как список, в котором можно выбрать по какому типу процентов будет происходить вычисление.

**Код программы**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, ExtCtrls;

type

{ TForm1 }

TForm1 = class(TForm)

btnCalculate: TButton;

cmbShape: TComboBox;

edtParam1: TEdit;

edtParam2: TEdit;

edtResult: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

lblParam1: TLabel;

lblParam2: TLabel;

Panel1: TPanel;

procedure btnCalculateClick(Sender: TObject);

procedure cmbShapeChange(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

procedure UpdateLabels;

public

end;

var

Form1: TForm1;

implementation

{$R \*.lfm}

{ TForm1 }

procedure TForm1.FormCreate(Sender: TObject);

begin

// Заполняем список фигур

cmbShape.Items.Add('Круг');

cmbShape.Items.Add('Прямоугольник');

cmbShape.Items.Add('Треугольник');

cmbShape.Items.Add('Трапеция');

cmbShape.ItemIndex := 0;

UpdateLabels;

end;

procedure TForm1.cmbShapeChange(Sender: TObject);

begin

UpdateLabels;

end;

procedure TForm1.btnCalculateClick(Sender: TObject);

var

area: Double;

param1, param2: Double;

begin

try

param1 := StrToFloat(edtParam1.Text);

// Для круга нужен только один параметр

if cmbShape.ItemIndex <> 0 then

param2 := StrToFloat(edtParam2.Text);

case cmbShape.ItemIndex of

0: area := Pi \* param1 \* param1; // Круг: π\*r²

1: area := param1 \* param2; // Прямоугольник: a\*b

2: area := 0.5 \* param1 \* param2; // Треугольник: 0.5\*a\*h

3: area := 0.5 \* (param1 + param2) \* param1; // Трапеция: 0.5\*(a+b)\*h (здесь param2 - второе основание)

end;

edtResult.Text := Format('%.2f', [area]);

except

on E: Exception do

ShowMessage('Ошибка ввода данных: ' + E.Message);

end;

end;

procedure TForm1.UpdateLabels;

begin

case cmbShape.ItemIndex of

0: begin // Круг

lblParam1.Caption := 'Радиус:';

lblParam2.Visible := False;

edtParam2.Visible := False;

end;

1: begin // Прямоугольник

lblParam1.Caption := 'Длина:';

lblParam2.Caption := 'Ширина:';

lblParam2.Visible := True;

edtParam2.Visible := True;

end;

2: begin // Треугольник

lblParam1.Caption := 'Основание:';

lblParam2.Caption := 'Высота:';

lblParam2.Visible := True;

edtParam2.Visible := True;

end;

3: begin // Трапеция

lblParam1.Caption := 'Высота:';

lblParam2.Caption := 'Второе основание:';

lblParam2.Visible := True;

edtParam2.Visible := True;

end;

end;

end;

end.

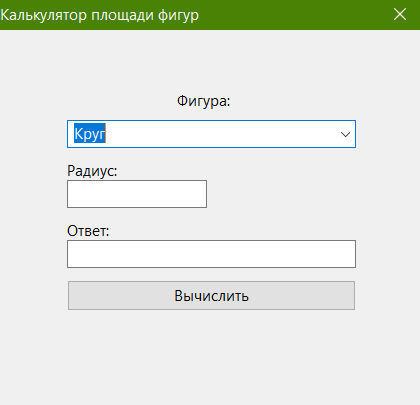


Рисунок 1 - Результат работы программы

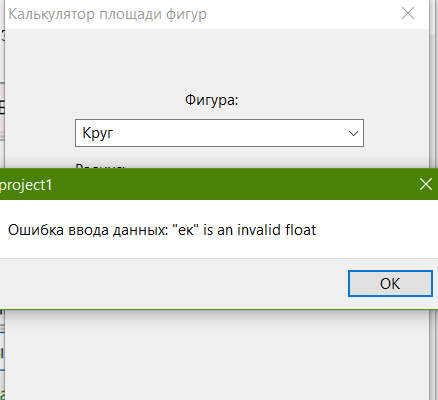


Рисунок 2 - Результат работы программы 1

**Вывод:** в ходе данной работы были получены навыки создания простых приложений в lazarus: был реализован калькулятор площади фигур. Также калькулятор выводит ошибку, если одно из полей не было заполнено, соответственно при вводе всех переменных вычисляет массу и выводит на экран.