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 ДИСЦИПЛИНЕ «Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-206-52-00

Долинин Владислав Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

В отчете должны отображаться:

1. Цель работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма и ответы на вопросы
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод
8. Цель: получение базовых навыков реализации приложений с графическим интерфейсом пользователя на основе событийно-ориентированной парадигмы.
9. Формулировка: реализовать приложения “Калькулятор объёма фигур”

Вариант 8

1. В моей программе используются 5 компонентов, а именно:

TLabel - служит для отображения текста на экране.

TEdit - стандартный управляющий элемент Windows для ввода. Он может быть использован для отображения короткого фрагмента текста и позволяет пользователю вводить текст во время выполнения программы.

TButton позволяет выполнить какие-либо действия при нажатии кнопки во время выполнения программы.

TBitBtn - кнопка вроде TButton, однако на ней можно разместить картинку (glyph). TBitBtn имеет несколько предопределенных типов (bkClose, bkOK и др), при выборе которых кнопка принимает соответствующий вид.

TImage - отображает графическое изображение на форме. Воспринимает форматы BMP, ICO, WMF. Если картинку подключить во время дизайна программы, то она прикомпилируется к EXE файлу.

1. Инициализация приложения

При запуске программы создается форма TForm1 с элементами:

3 поля ввода (Edit1, Edit2, Edit3)

4 кнопки (Button1-Button4 для операций +, -, \*, /)

2. Ввод данных

Пользователь вводит числа в Edit1 и Edit2

3. Выполнение операций

Кнопка "+" (Button1Click)

1. Получает числа из Edit1 и Edit2

2. Складывает их (Num1 + Num2)

3. Выводит результат в Edit3

Кнопка "-" (Button2Click)

1. Получает числа из Edit1 и Edit2

2. Вычитает (Num1 - Num2)

3. Выводит результат в Edit3

Кнопка "\*" (Button3Click)

1. Получает числа из Edit1 и Edit2

2. Умножает (Num1 \* Num2)

3. Выводит результат в Edit3

Кнопка "/" (Button4Click)

1. Получает числа из Edit1 и Edit2

2. Делит (Num1 / Num2)

3. Выводит результат в Edit3

Код программы:

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, Math; // Добавляем Math

type

TForm1 = class(TForm)

ButtonCalculate: TButton;

ComboBoxFigures: TComboBox;

EditParam1: TEdit;

EditParam2: TEdit;

LabelParam1: TLabel;

LabelParam2: TLabel;

LabelResult: TLabel;

LabelResultCaption: TLabel;

procedure ButtonCalculateClick(Sender: TObject);

procedure ComboBoxFiguresChange(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

procedure UpdateParameterFields;

public

end;

var

Form1: TForm1;

implementation

{$R \*.lfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

ComboBoxFigures.Items.Add('Куб');

ComboBoxFigures.Items.Add('Сфера');

ComboBoxFigures.Items.Add('Цилиндр');

ComboBoxFigures.Items.Add('Конус');

ComboBoxFigures.ItemIndex := 0;

UpdateParameterFields;

end;

procedure TForm1.ComboBoxFiguresChange(Sender: TObject);

begin

UpdateParameterFields;

end;

procedure TForm1.UpdateParameterFields;

begin

case ComboBoxFigures.ItemIndex of

0: // Куб

begin

LabelParam1.Caption := 'Сторона:';

LabelParam2.Visible := False;

EditParam2.Visible := False;

end;

1: // Сфера

begin

LabelParam1.Caption := 'Радиус:';

LabelParam2.Visible := False;

EditParam2.Visible := False;

end;

2, 3: // Цилиндр и Конус

begin

LabelParam1.Caption := 'Радиус:';

LabelParam2.Caption := 'Высота:';

LabelParam2.Visible := True;

EditParam2.Visible := True;

end;

end;

EditParam1.Text := '';

EditParam2.Text := '';

LabelResult.Caption := '';

end;

procedure TForm1.ButtonCalculateClick(Sender: TObject);

var

Param1, Param2, Volume: Double;

begin

try

Param1 := StrToFloat(EditParam1.Text);

if (ComboBoxFigures.ItemIndex in [2, 3]) then

Param2 := StrToFloat(EditParam2.Text)

else

Param2 := 0;

case ComboBoxFigures.ItemIndex of

0: Volume := Power(Param1, 3); // Куб: V = a^3

1: Volume := (4/3) \* Pi \* Power(Param1, 3); // Сфера: V = (4/3)πr^3

2: Volume := Pi \* Power(Param1, 2) \* Param2; // Цилиндр: V = πr^2h

3: Volume := (1/3) \* Pi \* Power(Param1, 2) \* Param2; // Конус: V = (1/3)πr^2h

end;

LabelResult.Caption := FormatFloat('0.##', Volume);

except

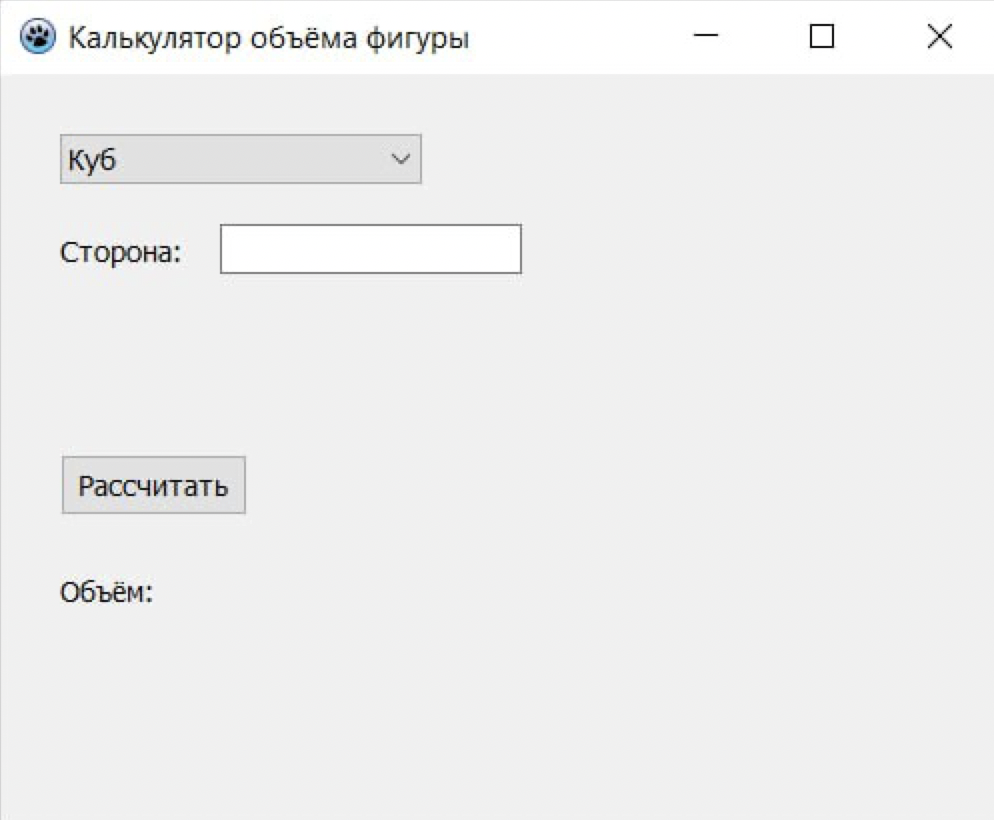
LabelResult.Caption := 'Ошибка ввода!';

end;

end;

end.

Результат выполнения:



Вывод: Получил базовые навыки реализации приложений с графическим интерфейсом пользователя на основе событийно-ориентированной парадигмы. При работе у меня возникло несколько проблем, при написании кода, а именно исправление вывод итога, какой должен быть объём, вывод в “TEdit”.