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 Казанцев Андрей Игоревич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

Задание:

Вариант 10

1. Реализовать приложение "Калькулятор площади фигур".

2. Предусмотреть выбор из фиксированного набора заданных фигур (не менее четырех).

**Ход работы:** с помощью данной программы пользователь может вводить данные для 4 разных фигур: квадрат, прямоугольник, треугольник и круг. После того, как пользователь введет данные, он может нажать на кнопку «рассчитать», после чего программа покажет сообщение, в котором будет написана площадь данной фигуры.

Форма представлена на рисунке номер 1:

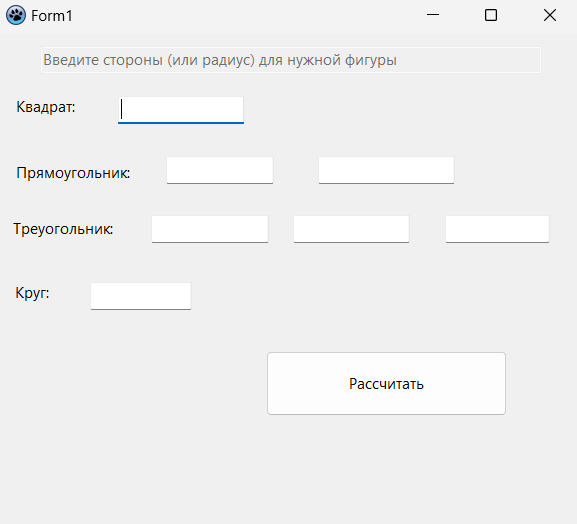


Рисунок 1 – форма программы.

Если пользователь введет нечисловое значение в одно из полей, программа выведет следующее сообщение:

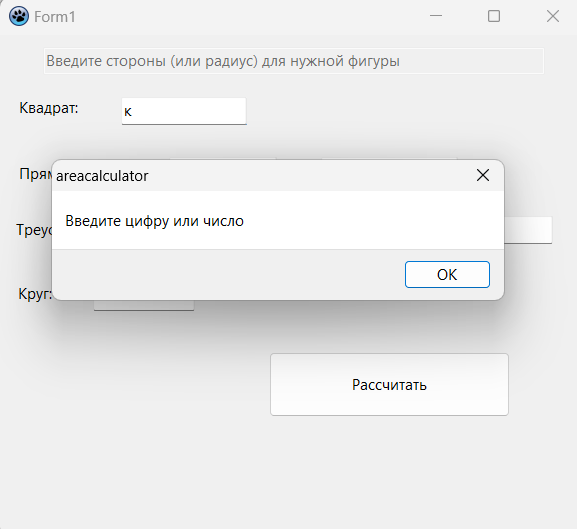


Рисунок 2 – сообщение о неправильности ввода

Если пользователь введет данные сразу для нескольких фигур, то программа выведет следующее сообщение:

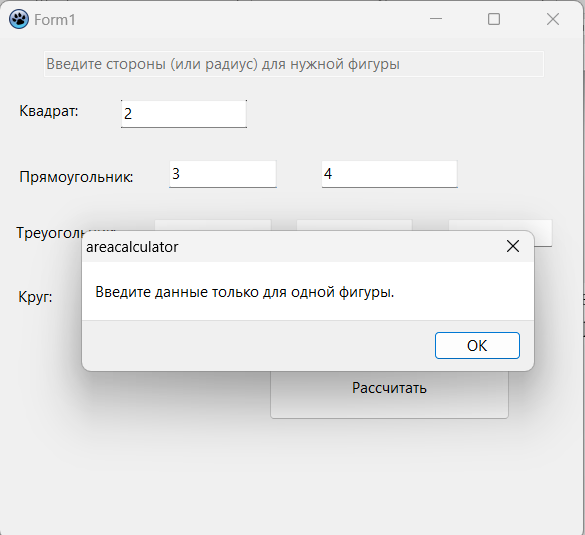


Рисунок 3 – сообщение при вводе данных для нескольких фигур

Программа также считывает дробные значения.

Результат выполнения программы представлен на рисунках 4-7:

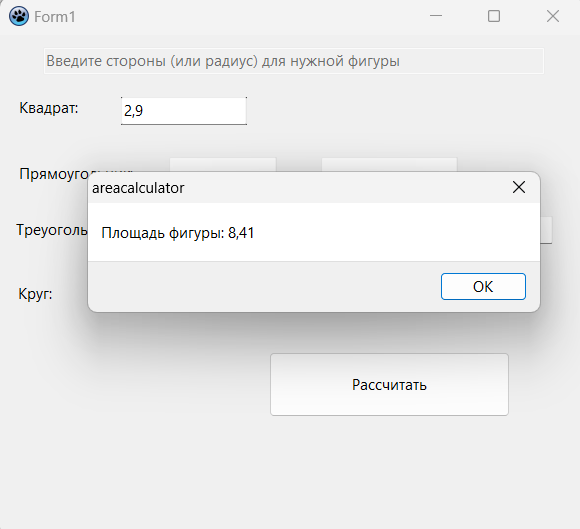


Рисунок 4 – площадь квадрата.

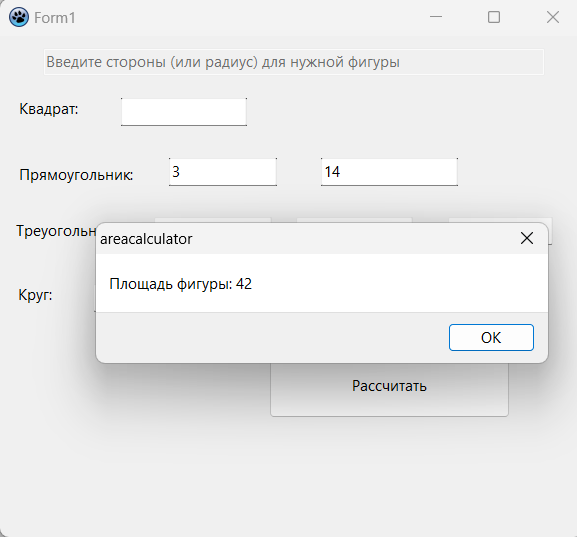


Рисунок 5 – площадь прямоугольника

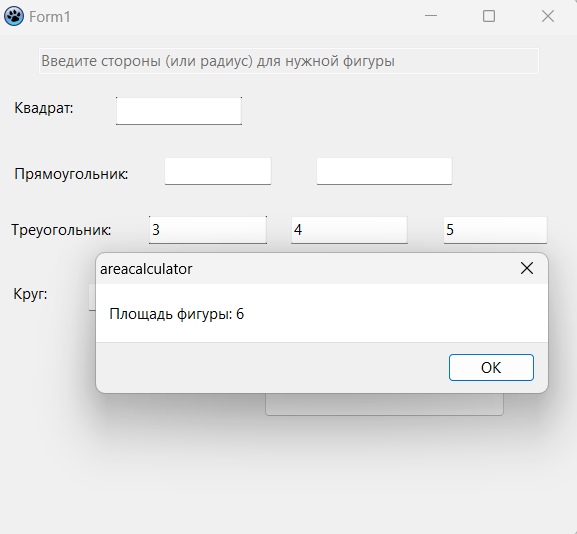


Рисунок 6 – площадь треугольника

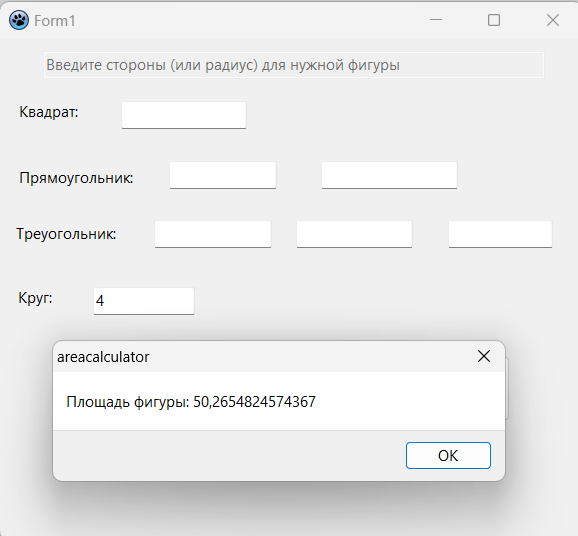


Рисунок 7 – площадь круга

**Код программы:**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

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

type

{ TForm1 }

TForm1 = class(TForm)

Button1: TButton;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

Edit6: TEdit;

Edit7: TEdit;

Edit8: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

procedure Button1Click(Sender: TObject);

procedure Edit1Change(Sender: TObject);

procedure Edit2Change(Sender: TObject);

procedure Edit3Change(Sender: TObject);

private

public

end;

var

Form1: TForm1;

implementation

{$R \*.lfm}

{ TForm1 }

procedure TForm1.Edit1Change(Sender: TObject);

begin

end;

procedure TForm1.Button1Click(Sender: TObject);

var

a, b, c, r, p, area: Double;

inputCount: Integer;

inputError: Boolean;

begin

inputCount := 0;

inputError := False;

area := 0;

// Проверка и расчет для квадрата

if Trim(Edit2.Text) <> '' then

begin

Inc(inputCount);

try

a := StrToFloat(Edit2.Text);

area := a \* a;

except

on E: EConvertError do

begin

inputError := True;

ShowMessage('Введите цифру или число');

end;

end;

end;

// Проверка и расчет для прямоугольника

if (Trim(Edit3.Text) <> '') and (Trim(Edit4.Text) <> '') then

begin

Inc(inputCount);

try

a := StrToFloat(Edit3.Text);

b := StrToFloat(Edit4.Text);

area := a \* b;

except

on E: EConvertError do

begin

inputError := True;

ShowMessage('Введите цифру или число');

end;

end;

end;

// Проверка и расчет для треугольника

if (Trim(Edit5.Text) <> '') and (Trim(Edit6.Text) <> '') and (Trim(Edit7.Text) <> '') then

begin

Inc(inputCount);

try

a := StrToFloat(Edit5.Text);

b := StrToFloat(Edit6.Text);

c := StrToFloat(Edit7.Text);

p := (a + b + c) / 2;

area := Sqrt(p \* (p - a) \* (p - b) \* (p - c));

except

on E: EConvertError do

begin

inputError := True;

ShowMessage('Введите цифру или число');

end;

end;

end;

// Проверка и расчет для круга

if Trim(Edit8.Text) <> '' then

begin

Inc(inputCount);

try

r := StrToFloat(Edit8.Text);

area := Pi \* r \* r;

except

on E: EConvertError do

begin

inputError := True;

ShowMessage('Введите цифру или число');

end;

end;

end;

// Проверка на количество введенных данных и наличие ошибок ввода

if inputError then

begin

// Сообщение об ошибке уже было показано

end

else if inputCount > 1 then

begin

ShowMessage('Введите данные только для одной фигуры.');

end

else if inputCount = 0 then

begin

ShowMessage('Данные не введены. Пожалуйста, введите данные для расчета площади.');

end

else

begin

// Вывод сообщения о площади фигуры, если нет ошибок и введены данные только для одной фигуры

if area > 0 then

ShowMessage('Площадь фигуры: ' + FloatToStr(area));

end;

end;

procedure TForm1.Edit2Change(Sender: TObject);

begin

end;

procedure TForm1.Edit3Change(Sender: TObject);

begin

end;

end.

**Вывод**

Цель данной работы заключалась в получении базовых навыков реализации приложений с графическим интерфейсом пользователя на основе событийно-ориентированной парадигмы. Эта цель была осуществима с помощью выполнения задания на создание калькулятора площади фигур. Данное задание отличалось количеством используемых процедур и функций. Тем самым оно вызывало больший интерес, т.к., выполняя, эти задания можно было столкнуться с неизвестными ранее функциями и процедурами в языке программирования Паскаль, такими как: StrToFloat, InputCount, InputError и др. Данные функции были изучены и применены в разработке программы для нахождения значения функции.

Не менее интересными были результаты выполнения программы. Площадь всех фигур высчитывалась с идеальной точностью, а проблемы с ошибочным вводом, например с вводом букв, решались соответствующим сообщением пользователю.

Подводя итог, можно сказать, что в данной работе была достигнута цель и были освоены новые способы выполнения заданий, а именно создание графического интерфейса. Работа проведена успешно, получен новый интересный и приятный опыт.