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»**

**«МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ»**

Выполнил: студент учебной группы

ИСПк-202-52-00

Долгополов Ярослав Максимович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

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

**Формулировка задания. Вариант 6**

**Постановка задачи**

1. Реализовать приложение "Калькулятор площади фигур".
2. Предусмотреть выбор из фиксированного набора заданных фигур (не менее четырех).

**Описание используемых элементов**

* Shapes: TShapes — форма, на которой размещены все объекты;
* SelectionOfShapes: TComboBox — список выбора плоской фигуры;
* SelectionOfShapesLabels: TLabel — надпись «Выберите фигуру:»;
* TriangleFormulas: TComboBox — список выбора способа подсчёта площади треугольника;
* TriangleFormulasLabel: TLabel — надпись «Выберите способ подсчёта:»;
* CalculationResult: TMemo — поле вывода результата;
* CalculationResultLabel: TLabel — надпись «Результаты:»;
* BorderResult: TShape — рамки для поля вывода результата;
* Entry1: TEdit — поле ввода первого числа;
* Entry2: TEdit — поле ввода второго числа;
* Entry3: TEdit — поле ввода третьего числа;
* Entry1Label: TLabel — пояснение для первого поля ввода;
* Entry2Label: TLabel — пояснение для второго поля ввода;
* Entry3Label: TLabel — пояснение для третьего поля ввода;
* BorderEntry1: TShape — рамки для первого поля ввода;
* BorderEntry2: TShape — рамки для второго поля ввода;
* BorderEntry3: TShape — рамки для третьего поля ввода;
* EntryAnswer: TLabel — стилизованная кнопка для подсчёта результата из введённых данных.

**Код программы**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

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

ExtCtrls, Buttons;

type

{ TShapes }

TShapes = class(TForm)

BorderEntry1: TShape;

BorderEntry2: TShape;

BorderEntry3: TShape;

BorderResult: TShape;

EntryAnswer: TLabel;

TriangleFormulasLabel: TLabel;

TriangleFormulas: TComboBox;

Entry1: TEdit;

Entry3Label: TLabel;

Entry3: TEdit;

Entry2: TEdit;

Entry1Label: TLabel;

Entry2Label: TLabel;

CalculationResultLabel: TLabel;

CalculationResult: TMemo;

SelectionOfShapesLabel: TLabel;

SelectionOfShapes: TComboBox;

procedure Entry1KeyPress(Sender: TObject; var Key: char);

procedure Entry2KeyPress(Sender: TObject; var Key: char);

procedure Entry3KeyPress(Sender: TObject; var Key: char);

procedure EntryAnswerClick(Sender: TObject);

procedure SelectionOfShapesChange(Sender: TObject);

procedure TriangleFormulasChange(Sender: TObject);

private

First, Second, Third: string;

public

end;

var

Shapes: TShapes;

implementation

{$R \*.lfm}

{ TShapes }

procedure TShapes.SelectionOfShapesChange(Sender: TObject);

begin

CalculationResult.Lines.Clear;

case SelectionOfShapes.Text of

'Параллелограмм (общий случай)':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := True;

Entry2.Visible := True;

Entry3Label.Visible := False;

Entry3.Visible := False;

BorderEntry1.Visible := True;

BorderEntry2.Visible := True;

BorderEntry3.Visible := False;

TriangleFormulas.Visible := False;

TriangleFormulasLabel.Visible := False;

Entry1Label.Caption := 'Введите длину высоты параллелограмма (в см):';

Entry2Label.Caption := 'Введите длину основания параллелограмма (в см):';

Entry1.Text := chr(0);

Entry2.Text := chr(0)

end;

'Квадрат':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := False;

Entry2.Visible := False;

Entry3Label.Visible := False;

Entry3.Visible := False;

BorderEntry1.Visible := True;

BorderEntry2.Visible := False;

BorderEntry3.Visible := False;

TriangleFormulas.Visible := False;

TriangleFormulasLabel.Visible := False;

Entry1Label.Caption := 'Введите длину стороны квадрата (в см):';

Entry1.Text := chr(0)

end;

'Прямоугольник':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := True;

Entry2.Visible := True;

Entry3Label.Visible := False;

Entry3.Visible := False;

BorderEntry1.Visible := True;

BorderEntry2.Visible := True;

BorderEntry3.Visible := False;

TriangleFormulas.Visible := False;

TriangleFormulasLabel.Visible := False;

Entry1Label.Caption := 'Введите длину прямоугольника (в см):';

Entry2Label.Caption := 'Введите ширину прямоугольника (в см):';

Entry1.Text := chr(0);

Entry2.Text := chr(0)

end;

'Ромб':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := True;

Entry2.Visible := True;

Entry3Label.Visible := False;

Entry3.Visible := False;

BorderEntry1.Visible := True;

BorderEntry2.Visible := True;

BorderEntry3.Visible := False;

TriangleFormulas.Visible := False;

TriangleFormulasLabel.Visible := False;

Entry1Label.Caption := 'Введите длину первой диагонали (в см):';

Entry2Label.Caption := 'Введите длину второй диагонали (в см):';

Entry1.Text := chr(0);

Entry2.Text := chr(0)

end;

'Круг':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := False;

Entry2.Visible := False;

Entry3Label.Visible := False;

Entry3.Visible := False;

BorderEntry1.Visible := True;

BorderEntry2.Visible := False;

BorderEntry3.Visible := False;

TriangleFormulas.Visible := False;

TriangleFormulasLabel.Visible := False;

Entry1Label.Caption := 'Введите радиус круга (в см):';

Entry1.Text := chr(0)

end;

'Трапеция':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := True;

Entry2.Visible := True;

Entry3Label.Visible := True;

Entry3.Visible := True;

BorderEntry1.Visible := True;

BorderEntry2.Visible := True;

BorderEntry3.Visible := True;

TriangleFormulas.Visible := False;

TriangleFormulasLabel.Visible := False;

Entry1Label.Caption := 'Введите длину первого основания (в см):';

Entry2Label.Caption := 'Введите длину второго основания (в см):';

Entry3Label.Caption := 'Введите длину высоты трапеции (в см):';

Entry1.Text := chr(0);

Entry2.Text := chr(0);

Entry3.Text := chr(0)

end;

'Треугольник':

begin

Entry1Label.Visible := False;

Entry1.Visible := False;

Entry2Label.Visible := False;

Entry2.Visible := False;

Entry3Label.Visible := False;

Entry3.Visible := False;

BorderEntry1.Visible := False;

BorderEntry2.Visible := False;

BorderEntry3.Visible := False;

TriangleFormulas.ItemIndex := -1;

TriangleFormulas.Visible := True;

TriangleFormulasLabel.Visible := True;

end;

'Правильный многоугольник':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := True;

Entry2.Visible := True;

Entry3Label.Visible := False;

Entry3.Visible := False;

BorderEntry1.Visible := True;

BorderEntry2.Visible := True;

BorderEntry3.Visible := False;

TriangleFormulas.Visible := False;

TriangleFormulasLabel.Visible := False;

Entry1Label.Caption := 'Введите количество сторон:';

Entry2Label.Caption := 'Введите длину стороны многоугольника (в см):';

Entry1.Text := chr(0);

Entry2.Text := chr(0)

end;

end;

end;

procedure TShapes.TriangleFormulasChange(Sender: TObject);

begin

CalculationResult.Lines.Clear;

case TriangleFormulas.Text of

'По трём сторонам':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := True;

Entry2.Visible := True;

Entry3Label.Visible := True;

Entry3.Visible := True;

BorderEntry1.Visible := True;

BorderEntry2.Visible := True;

BorderEntry3.Visible := True;

Entry1Label.Caption := 'Введите длину первой стороны (в см):';

Entry2Label.Caption := 'Введите длину второй стороны (в см):';

Entry3Label.Caption := 'Введите длину третьей стороны (в см):';

Entry1.Text := chr(0);

Entry2.Text := chr(0);

Entry3.Text := chr(0)

end;

'По двум сторонам и углу между ними':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := True;

Entry2.Visible := True;

Entry3Label.Visible := True;

Entry3.Visible := True;

BorderEntry1.Visible := True;

BorderEntry2.Visible := True;

BorderEntry3.Visible := True;

Entry1Label.Caption := 'Введите длину первой стороны (в см):';

Entry2Label.Caption := 'Введите длину второй стороны (в см):';

Entry3Label.Caption := 'Введите размер угла между ними (в градусах):';

Entry1.Text := chr(0);

Entry2.Text := chr(0);

Entry3.Text := chr(0)

end;

'По стороне и двум прилежащим углам':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := True;

Entry2.Visible := True;

Entry3Label.Visible := True;

Entry3.Visible := True;

BorderEntry1.Visible := True;

BorderEntry2.Visible := True;

BorderEntry3.Visible := True;

Entry1Label.Caption := 'Введите длину стороны (в см):';

Entry2Label.Caption := 'Введите размер первого угла (в градусах):';

Entry3Label.Caption := 'Введите размер второго угла (в градусах):';

Entry1.Text := chr(0);

Entry2.Text := chr(0);

Entry3.Text := chr(0)

end;

'По основанию и высоте':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := True;

Entry2.Visible := True;

Entry3Label.Visible := False;

Entry3.Visible := False;

BorderEntry1.Visible := True;

BorderEntry2.Visible := True;

BorderEntry3.Visible := False;

Entry1Label.Caption := 'Введите длину основания (в см):';

Entry2Label.Caption := 'Введите длину высоты (в см):';

Entry1.Text := chr(0);

Entry2.Text := chr(0)

end;

'По двум катетам':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := True;

Entry2.Visible := True;

Entry3Label.Visible := False;

Entry3.Visible := False;

BorderEntry1.Visible := True;

BorderEntry2.Visible := True;

BorderEntry3.Visible := False;

Entry1Label.Caption := 'Введите длину первого катета (в см):';

Entry2Label.Caption := 'Введите длину второго катета (в см):';

Entry1.Text := chr(0);

Entry2.Text := chr(0)

end;

'По гипотенузе и острому углу':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := True;

Entry2.Visible := True;

Entry3Label.Visible := False;

Entry3.Visible := False;

BorderEntry1.Visible := True;

BorderEntry2.Visible := True;

BorderEntry3.Visible := False;

Entry1Label.Caption := 'Введите длину гипотенузы (в см):';

Entry2Label.Caption := 'Введите размер острого угла (в градусах):';

Entry1.Text := chr(0);

Entry2.Text := chr(0)

end;

'По катету и прилежащему острому углу':

begin

Entry1Label.Visible := True;

Entry1.Visible := True;

Entry2Label.Visible := True;

Entry2.Visible := True;

Entry3Label.Visible := False;

Entry3.Visible := False;

BorderEntry1.Visible := True;

BorderEntry2.Visible := True;

BorderEntry3.Visible := False;

Entry1Label.Caption := 'Введите длину катета (в см):';

Entry2Label.Caption := 'Введите размер острого угла (в градусах):';

Entry1.Text := chr(0);

Entry2.Text := chr(0)

end;

end;

end;

procedure TShapes.Entry1KeyPress(Sender: TObject; var Key: char);

begin

Entry1.ReadOnly:= not ((Key in ['0'..'9', #8]) or ((Key in [',','.']) and ((pos(',', Entry1.Text) = 0) and (pos('.', Entry1.Text) = 0)) and (Entry1.Text <> '')))

end;

procedure TShapes.Entry2KeyPress(Sender: TObject; var Key: char);

begin

Entry2.ReadOnly:= not ((Key in ['0'..'9',#8]) or ((Key in [',','.']) and ((pos(',', Entry2.Text) = 0) and (pos('.', Entry2.Text) = 0)) and (Entry2.Text <> '')))

end;

procedure TShapes.Entry3KeyPress(Sender: TObject; var Key: char);

begin

Entry3.ReadOnly:= not ((Key in ['0'..'9',#8]) or ((Key in [',','.']) and ((pos(',', Entry3.Text) = 0) and (pos('.', Entry3.Text) = 0)) and (Entry3.Text <> '')))

end;

procedure TShapes.EntryAnswerClick(Sender: TObject);

var AnswerNum, p, FirstNum, SecondNum, ThirdNum: real;

t: boolean;

begin

CalculationResult.Lines.Clear;

t:=False;

case SelectionOfShapes.Text of

'Параллелограмм (общий случай)':

begin

First:=Entry1.Text;

Second:=Entry2.Text;

if (First = '') or (Second = '') or (StrToFloat(First) = 0) or (StrToFloat(Second) = 0) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой и параметр не должен быть равен 0.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

SecondNum:=(Round(StrToFloat(Second)\*100)/100);

AnswerNum:=FirstNum \* SecondNum;

end;

end;

'Квадрат':

begin

First:=Entry1.Text;

if (First = '') or (StrToFloat(First) = 0) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой и параметр не должен быть равен 0.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

AnswerNum:=sqr(FirstNum);

end;

end;

'Прямоугольник':

begin

First:=Entry1.Text;

Second:=Entry2.Text;

if (First = '') or (Second = '') or (StrToFloat(First) = 0) or (StrToFloat(Second) = 0) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой и параметр не должен быть равен 0.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

SecondNum:=(Round(StrToFloat(Second)\*100)/100);

AnswerNum:=FirstNum \* SecondNum;

end;

end;

'Ромб':

begin

First:=Entry1.Text;

Second:=Entry2.Text;

if (First = '') or (Second = '') or (StrToFloat(First) = 0) or (StrToFloat(Second) = 0) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой и параметр не должен быть равен 0.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

SecondNum:=(Round(StrToFloat(Second)\*100)/100);

AnswerNum:=0.5 \* FirstNum \* SecondNum;

end;

end;

'Круг':

begin

First:=Entry1.Text;

if (First = '') or (StrToFloat(First) = 0) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой и параметр не должен быть равен 0.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

AnswerNum:=Pi\*sqr(FirstNum);

end;

end;

'Трапеция':

begin

First:=Entry1.Text;

Second:=Entry2.Text;

Third:=Entry3.Text;

if (First = '') or (Second = '') or (Third = '') or (StrToFloat(First) = 0) or (StrToFloat(Second) = 0) or (StrToFloat(Third) = 0) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой и параметр не должен быть равен 0.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

SecondNum:=(Round(StrToFloat(Second)\*100)/100);

ThirdNum:=(Round(StrToFloat(Third)\*100)/100);

AnswerNum:=0.5 \* (FirstNum + SecondNum) \* ThirdNum;

end;

end;

'Треугольник':

begin

case TriangleFormulas.Text of

'По трём сторонам':

begin

First:=Entry1.Text;

Second:=Entry2.Text;

Third:=Entry3.Text;

if (First = '') or (Second = '') or (Third = '') or (StrToFloat(First) = 0) or (StrToFloat(Second) = 0) or (StrToFloat(Third) = 0) or (StrToFloat(Second)+StrToFloat(Third) <= StrToFloat(First)) or (StrToFloat(First)+StrToFloat(Third) <= StrToFloat(Second)) or (StrToFloat(First)+StrToFloat(Second) <= StrToFloat(Third)) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой, параметр не должен быть равен 0 и сумма двух сторон не должна быть больше третьей.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

SecondNum:=(Round(StrToFloat(Second)\*100)/100);

ThirdNum:=(Round(StrToFloat(Third)\*100)/100);

p:= (FirstNum+SecondNum+ThirdNum)/2;

AnswerNum:=sqrt(p\*(p - FirstNum)\*(p - SecondNum)\*(p - ThirdNum));

end;

end;

'По двум сторонам и углу между ними':

begin

First:=Entry1.Text;

Second:=Entry2.Text;

Third:=Entry3.Text;

if (First = '') or (Second = '') or (Third = '') or (StrToFloat(First) = 0) or (StrToFloat(Second) = 0) or (StrToFloat(Third) = 0) or (StrToFloat(Third)>=180) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой, параметр не должен быть равен 0 и размер угла треугольника не должен превышать 180 градусов.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

SecondNum:=(Round(StrToFloat(Second)\*100)/100);

ThirdNum:=(Round(StrToFloat(Third)\*100)/100);

AnswerNum:=0.5 \* FirstNum \* SecondNum \* sin(ThirdNum\*2\*Pi/360);

end;

end;

'По стороне и двум прилежащим углам':

begin

First:=Entry1.Text;

Second:=Entry2.Text;

Third:=Entry3.Text;

if (First = '') or (Second = '') or (Third = '') or (StrToFloat(First) = 0) or (StrToFloat(Second) = 0) or (StrToFloat(Third) = 0) or (StrToFloat(Third)>=180) or (StrToFloat(Second)>=180) or ((StrToFloat(Third)+StrToFloat(Second))>=180) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой и параметр не должен быть равен 0, размер угла треугольника и сумма двух углов должны быть меньше 180 градусов.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

SecondNum:=(Round(StrToFloat(Second)\*100)/100);

ThirdNum:=(Round(StrToFloat(Third)\*100)/100);

AnswerNum:= sqr(FirstNum)\*(sin(SecondNum\*2\*Pi/360)\*sin(ThirdNum\*2\*Pi/360))/2\*sin((ThirdNum\*2\*Pi/360)+(SecondNum\*2\*Pi/360));

end;

end;

'По основанию и высоте':

begin

First:=Entry1.Text;

Second:=Entry2.Text;

if (First = '') or (Second = '') or (StrToFloat(First) = 0) or (StrToFloat(Second) = 0) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой и параметр не должен быть равен 0.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

SecondNum:=(Round(StrToFloat(Second)\*100)/100);

AnswerNum:=0.5 \* FirstNum \* SecondNum;

end;

end;

'По двум катетам':

begin

First:=Entry1.Text;

Second:=Entry2.Text;

if (First = '') or (Second = '') or (StrToFloat(First) = 0) or (StrToFloat(Second) = 0) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой и параметр не должен быть равен 0.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

SecondNum:=(Round(StrToFloat(Second)\*100)/100);

AnswerNum:=0.5 \* FirstNum \* SecondNum;

end;

end;

'По гипотенузе и острому углу':

begin

First:=Entry1.Text;

Second:=Entry2.Text;

if (First = '') or (Second = '') or (StrToFloat(First) = 0) or (StrToFloat(Second) = 0) or (StrToFloat(Second)>=180) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой, параметр не должен быть равен 0 и размер угла треугольника не должен превышать 180 градусов.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

SecondNum:=(Round(StrToFloat(Second)\*100)/100);

AnswerNum:=0.25 \* sqr(FirstNum) \* sin(2\*SecondNum\*2\*Pi/360);

end;

end;

'По катету и прилежащему острому углу':

begin

First:=Entry1.Text;

Second:=Entry2.Text;

if (First = '') or (Second = '') or (StrToFloat(First) = 0) or (StrToFloat(Second) = 0) or (StrToFloat(Second)>=180) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой, параметр не должен быть равен 0 и размер угла треугольника не должен превышать 180 градусов.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

SecondNum:=(Round(StrToFloat(Second)\*100)/100);

AnswerNum:=0.5 \* sqr(FirstNum) \* tan(SecondNum\*2\*Pi/360);

end;

end;

end;

end;

'Правильный многоугольник':

begin

First:=Entry1.Text;

Second:=Entry2.Text;

if (First = '') or (Second = '') or (StrToFloat(First) = 0) or (StrToFloat(Second) = 0) or (StrToFloat(First)<>Round(StrToFloat(First))) then CalculationResult.Lines.Add('Ошибка! Неверный формат ввода! Строка ввода не должна быть пустой, параметр не должен быть равен 0 и количество сторон должно быть выражено целым числом.')

else

begin

t:=True;

FirstNum:=(Round(StrToFloat(First)\*100)/100);

SecondNum:=(Round(StrToFloat(Second)\*100)/100);

AnswerNum:=(FirstNum/4)\*sqr(SecondNum)\* Cot(Pi/FirstNum);

end;

end;

end;

if t then

begin

CalculationResult.Font.Size:=20;

CalculationResult.Lines.Add(FloatToStr(Round(AnswerNum\*100)/100) + ' кв.см')

end;

end;

end.

**Результат выполнения программы**

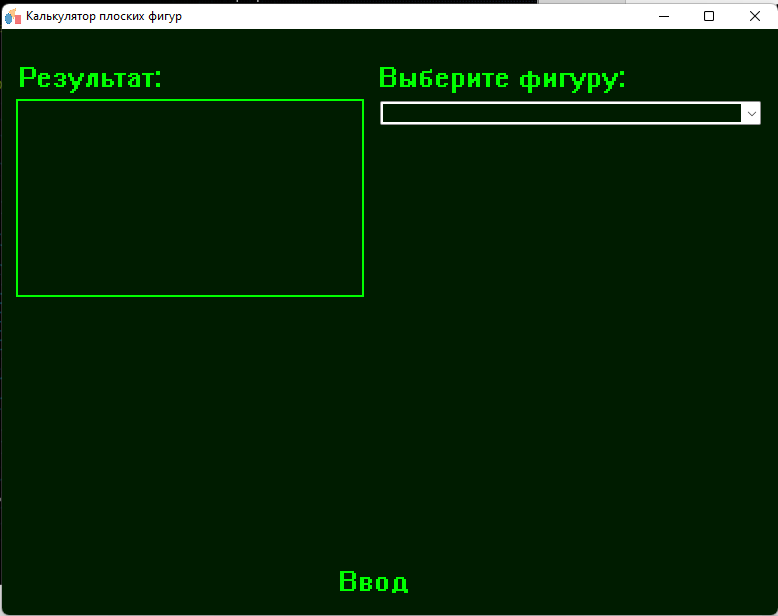
****

Рисунок 1 — интерфейс калькулятора

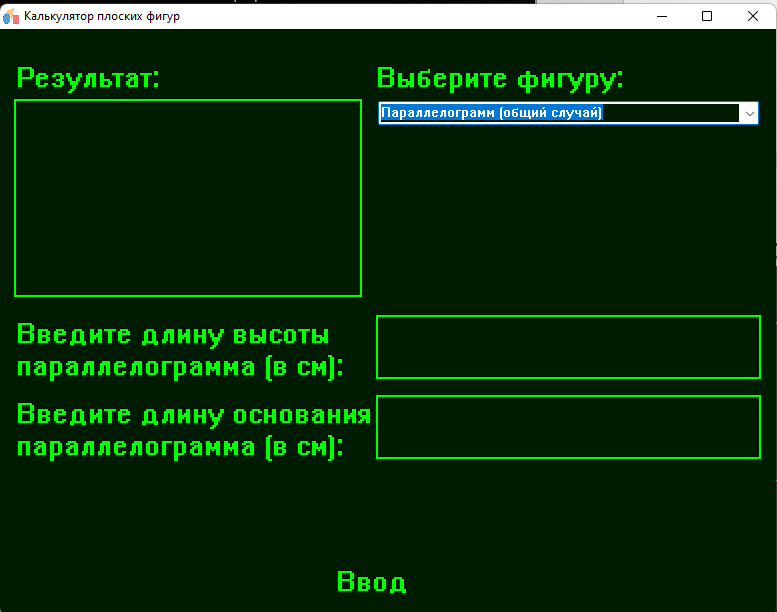
****

Рисунок 2 — выбор фигуры

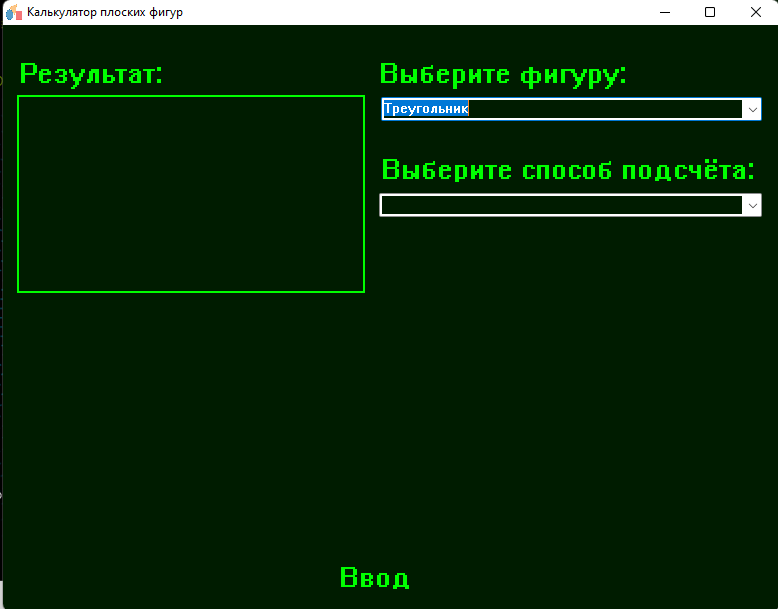
****

Рисунок 2 — выбор треугольник с выбором формулы

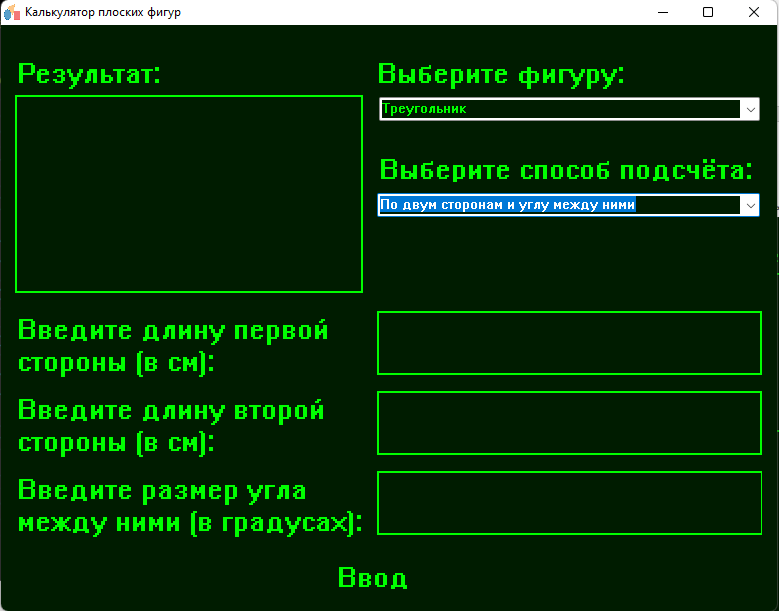
****

Рисунок 3 — выбором формулы

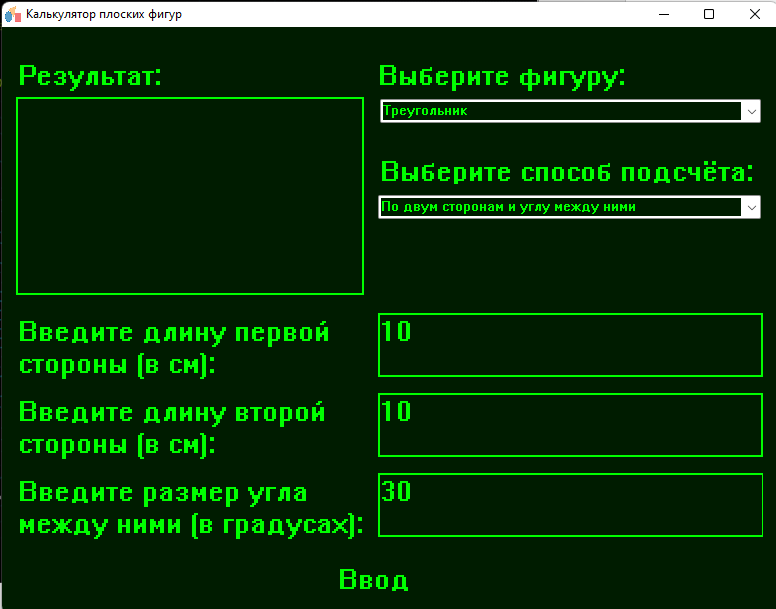
****

Рисунок 4 — ввод данных

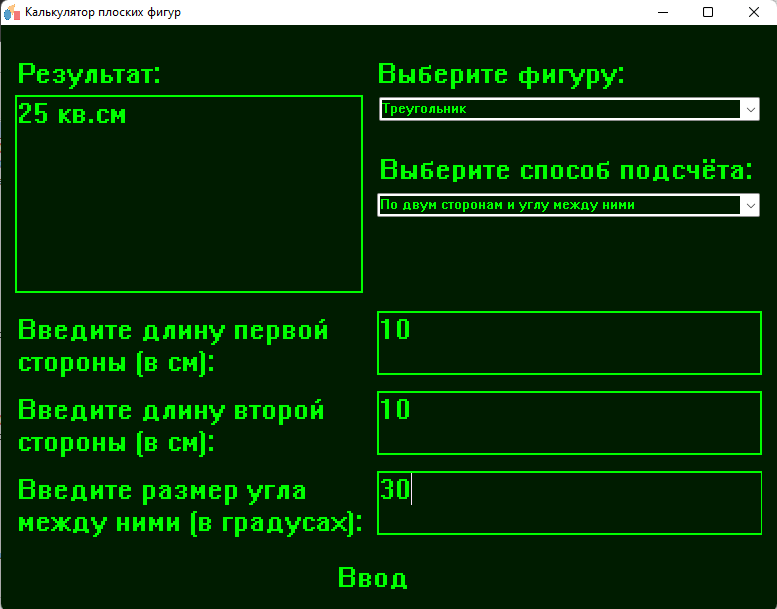
****

Рисунок 5 — вывод результата

**Вывод**

Во время этой работы было реализовано приложение «Калькулятор площади фигур», в которой было предусмотрен выбор из фиксированного набора заданных фигур (не менее четырех).

Калькулятор для каждой фигуры подбирает формулу, подставляя введённые параметры в неё, и выводит результат в отдельное поле.

Были получены базовые навыки реализации приложений с графическим интерфейсом пользователя на основе событийно-ориентированной парадигмы и увеличены познания в среде разработке Lazarus на основе Pascal.