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 «МДК 05.02 Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-206-52-00

Табак Евгений Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. Цель работы

Получение базовых навыков реализации приложений с графическим интерфейсом пользователя на основе событийно-ориентированной парадигмы.

1. Формулировка задания (с вариантом)

Вариант: 12

1. Реализовать приложение "Калькулятор объема фигур

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

**Описание алгоритма:**

Программа позволяет пользователю рассчитывать объём различных геометрических фигур: куба, сферы, цилиндра и конуса. Пользователь выбирает тип фигуры из выпадающего списка, вводит необходимые параметры, такие как радиус или высоту, и нажимает кнопку для выполнения расчёта. Результаты отображаются прямо на форме.

Процесс работы программы можно разделить на три основных шага:

1. **Выбор фигуры:**

При изменении выбора в выпадающем списке cbFigureType, метод cbFigureTypeChange адаптирует интерфейс, показывая или скрывая поля для ввода параметров. Это позволяет пользователю видеть только те поля, которые необходимы для расчёта выбранной фигуры.

1. **Расчёт объёма:**

Метод CalculateVolume выполняет основные расчёты. Перед расчётом значения из текстовых полей преобразуются в числа. Затем, в зависимости от выбранной фигуры, применяются следующие математические формулы:

Куб: объём вычисляется как куб стороны.

Сфера: объём сферы рассчитывается по формуле (4/3) \* π \* r^3.

Цилиндр: объём цилиндра определяется как π \* r^2 \* h.

Конус: объём конуса вычисляется по формуле (1/3) \* π \* r^2 \* h.

1. **Отображение результата:**

По нажатию кнопки btnCalculate, программа вызывает CalculateVolume, получает объём и отображает его на форме в метках lblResult, форматируя вывод до трёх десятичных знаков.

**Код программы:**

**unit Unit1;**

**{$mode objfpc}{$H+}**

**interface**

**uses**

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

**type**

**{ TNet }**

**TNet = class(TForm)**

**btnCalculate: TButton;**

**cbFigureType: TComboBox;**

**edParam1: TEdit;**

**edParam2: TEdit;**

**edParam3: TEdit;**

**Label1: TLabel;**

**lblResult: TLabel;**

**lblParam1: TLabel;**

**lblParam2: TLabel;**

**lblParam3: TLabel;**

**procedure btnCalculateClick(Sender: TObject);**

**procedure cbFigureTypeChange(Sender: TObject);**

**private**

**function CalculateVolume: Double;**

**public**

**end;**

**var**

**Net: TNet;**

**implementation**

**{$R \*.lfm}**

**{ TNet }**

**function TNet.CalculateVolume: Double;**

**var**

**Param1, Param2, Param3: Double;**

**begin**

**// Переводим текст в числовые значения, по умолчанию используем 0**

**Param1 := StrToFloatDef(edParam1.Text, 0);**

**Param2 := StrToFloatDef(edParam2.Text, 0);**

**Param3 := StrToFloatDef(edParam3.Text, 0);**

**// Расчёт объёма в зависимости от выбранной фигуры**

**case cbFigureType.ItemIndex of**

**0: Result := Power(Param1, 3); // Объём куба**

**1: Result := (4.0/3.0) \* Pi \* Power(Param1, 3); // Объём сферы**

**2: Result := Pi \* Power(Param1, 2) \* Param2; // Объём цилиндра**

**3: Result := (1.0/3.0) \* Pi \* Power(Param1, 2) \* Param2; // Объём конуса**

**else**

**Result := 0;**

**end;**

**end;**

**procedure TNet.cbFigureTypeChange(Sender: TObject);**

**begin**

**edParam1.Visible := False;**

**lblParam1.Visible := False;**

**edParam2.Visible := False;**

**lblParam2.Visible := False;**

**edParam3.Visible := False;**

**lblParam3.Visible := False;**

**// Определяем, какие поля ввода и метки показывать**

**case cbFigureType.ItemIndex of**

**0: begin // Куб**

**lblParam1.Caption := 'Длина стороны куба (м):';**

**edParam1.Visible := True;**

**lblParam1.Visible := True;**

**end;**

**1: begin // Сфера**

**lblParam1.Caption := 'Радиус сферы (м):';**

**edParam1.Visible := True;**

**lblParam1.Visible := True;**

**end;**

**2: begin // Цилиндр**

**lblParam1.Caption := 'Радиус цилиндра (м):';**

**lblParam2.Caption := 'Высота цилиндра (м):';**

**edParam1.Visible := True;**

**lblParam1.Visible := True;**

**edParam2.Visible := True;**

**lblParam2.Visible := True;**

**end;**

**3: begin // Конус**

**lblParam1.Caption := 'Радиус конуса (м):';**

**lblParam2.Caption := 'Высота конуса (м):';**

**edParam1.Visible := True;**

**lblParam1.Visible := True;**

**edParam2.Visible := True;**

**lblParam2.Visible := True;**

**end;**

**end;**

**end;**

**procedure TNet.btnCalculateClick(Sender: TObject);**

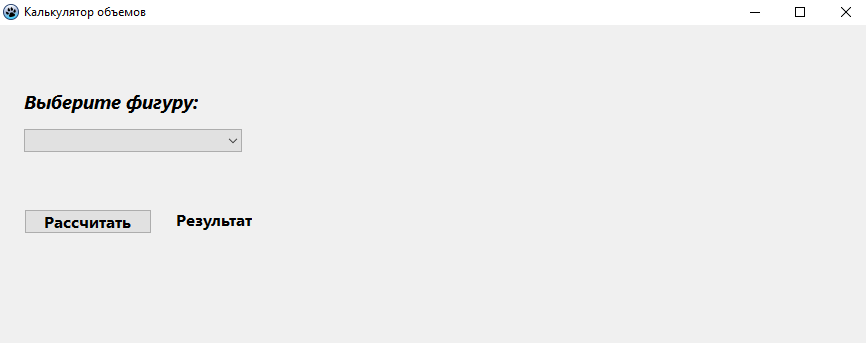
**begin**

**lblResult.Caption := 'Объём: ' + FloatToStrF(CalculateVolume, ffFixed, 8, 3) + ' м³';**

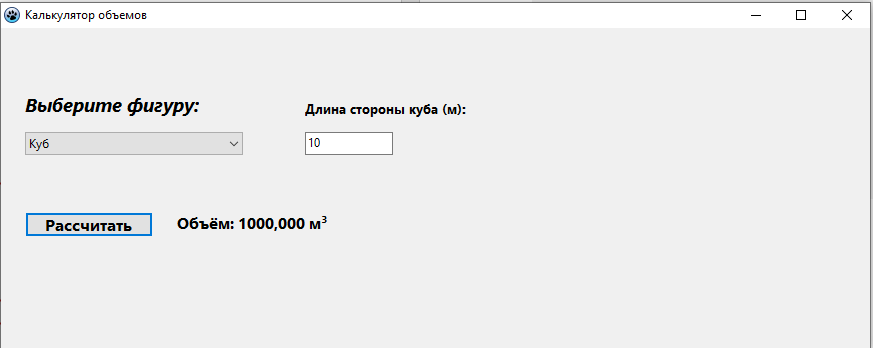
**end;**

**end.**

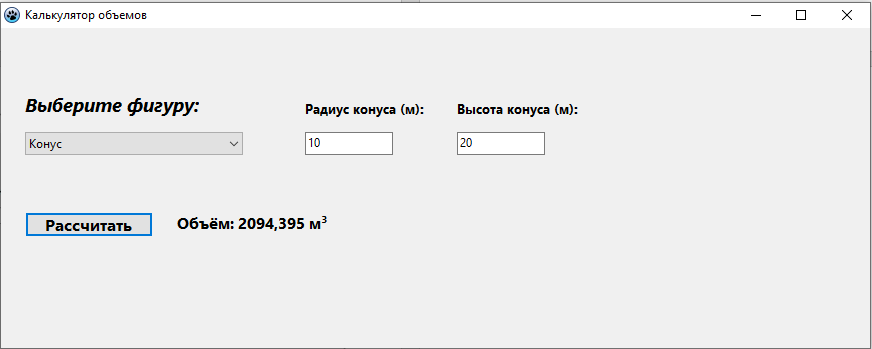
**Результат выполнения программы:**

****

Пример работы программы №1 – Рис.1



Пример работы программы №2 – Рис.2



Пример работы программы №3 – Рис.3

**Вывод**

В данной работе была достигнута цель по получению базовых навыков в реализации приложений с графическим интерфейсом пользователя.

Разработанный "Калькулятор объема фигур" позволяет вычислять объемы куба, сферы, цилиндра и конуса с динамическим изменением интерфейса в зависимости от выбора пользователя. Это приложение демонстрирует интеграцию графического интерфейса с математическими функциями и обеспечивает практический опыт в программировании GUI, что способствует углублению технических знаний и умений в разработке программного обеспечения.