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-204-52-00

Вейс Роман Михайлович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. Цель работы

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

2. Формулировка задания.

Вариант 6.

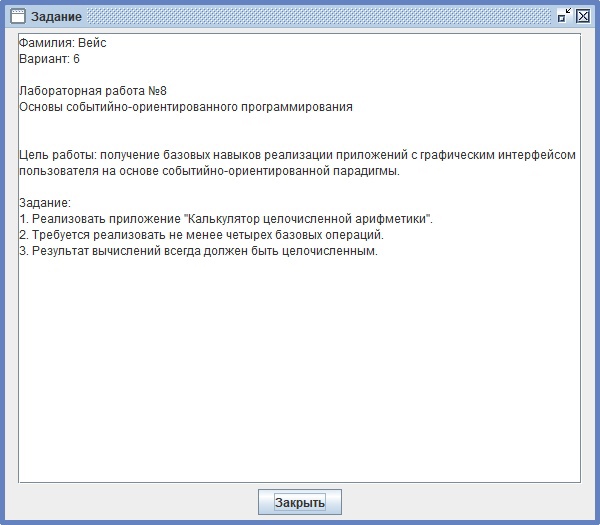


Рисунок 1 – Вариант работы.

3. Описание алгоритма.

Данный алгоритм предназначен для отрисовки калькулятора целочисленной арифметики.

1. Программа ожидает ввода первого числа с помощью кнопок цифр.

2. Пользователь выбирает операцию с помощью соответствующей кнопки.

3. Программа ожидает ввода второго числа.

4. При нажатии кнопки "=" программа выполняет выбранную операцию над введёнными числами.

5. Результат операции отображается в поле ввода.

6. Программа готова к новому вычислению.

4. Код программы.

Задание 1.

Модуль.

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

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

type

{ TForm1 }

TForm1 = class(TForm)

ButtonUmnoz: TButton;

ButtonMinus: TButton;

ButtonPlus: TButton;

Button5: TButton;

Button4: TButton;

Button1: TButton;

Button6: TButton;

Button2: TButton;

Button3: TButton;

Button0: TButton;

ButtonDiv: TButton;

ButtonEqual: TButton;

ButtonClear: TButton;

ButtonClearEntry: TButton;

Button7: TButton;

ButtonLastClear: TButton;

ButtonMod: TButton;

ClearButton: TButton;

Button9: TButton;

Button8: TButton;

Edit1: TEdit;

procedure ClearEntryClick(Sender: TObject);

procedure Clear\_LastClick(Sender: TObject);

procedure ClearClick(Sender: TObject);

procedure ClickZnak(Sender: TObject);

procedure ClickBut(Sender: TObject);

procedure Edit1Change(Sender: TObject);

procedure equalClick(Sender: TObject);

private

public

end;

var

Form1: TForm1;

a, b, c: real;

d, e, f: integer;

znak : string;

implementation

{$R \*.lfm}

{ TForm1 }

procedure TForm1.ClearEntryClick(Sender: TObject);

begin

Edit1.Clear;

end;

procedure TForm1.Clear\_LastClick(Sender: TObject);

var

str: string;

begin

str:= Edit1.Text;

if str <> '' then

delete(str, Length(str),1);

Edit1.Text:=str;

end;

procedure TForm1.ClearClick(Sender: TObject);

begin

Edit1.Clear;

a:=0;

b:=0;

c:=0;

d:=0;

e:=0;

f:=0;

end;

procedure TForm1.ClickZnak(Sender: TObject);

begin

if Edit1.Text = '' then

ShowMessage('Введите число')

else

begin

d:= strToInt(Edit1.Text);

a:= strToFloat(Edit1.Text);

Edit1.clear;

znak:=(Sender as TButton).Caption;

end;

end;

procedure TForm1.ClickBut(Sender: TObject);

begin

Edit1.Text:=Edit1.Text + (Sender as TButton).Caption;

end;

procedure TForm1.Edit1Change(Sender: TObject);

begin

end;

procedure TForm1.equalClick(Sender: TObject);

begin

if Edit1.Text = '' then

ShowMessage('Ввидите число')

else

begin

e:= strToInt(Edit1.text);

b:= strtoFLoat(Edit1.Text);

Edit1.Clear;

case znak of

'+' : c:= a+b;

'-' : c:= a-b;

'\*' : c:=a\*b;

'/': if b = 0 then showmessage('На ноль делить нельзая') else

c:=a/b;

'Div': begin

f:=d div e;

c:=f;

end;

'Mod':begin

f:=d mod e;

c:=f;

end;

end;

end;

Edit1.Text:= FloatToStr(f);

Edit1.Text:= FloatToStr(c);

end;

end.

5. Вывод.

Выполняя домашнюю контрольную работу, мы смогли достичь поставленной цели: узнали, как работать с приложениями с графическим интерфейсом пользователя на основе событийно-ориентированной парадигмы.

В ходе работы над задание мы изучили принципы работы с IDLE Lazarus.