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

Криницын Олег Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель домашней контрольной работы:**

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

1. **Скриншот формулировки задания**

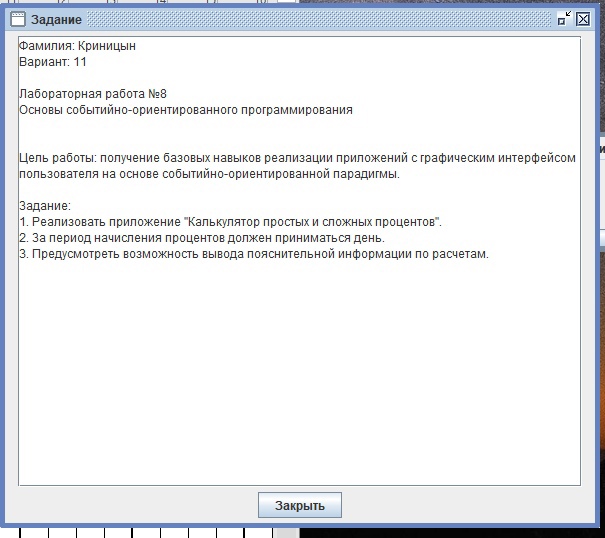


Рисунок 1 - Формулировка задания

1. **Код программы**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, ExtCtrls,

Buttons, Math;

type

{ TForm1 }

TForm1 = class(TForm)

btnClose: TBitBtn;

btnCalculate: TButton;

cbxCalculationType: TComboBox;

edtPrincipal: TEdit;

edtRate: TEdit;

edtTime: TEdit;

lblProcent: TLabel;

lblPrincipal: TLabel;

lblRate: TLabel;

lblTime: TLabel;

memResult: TMemo;

procedure btnCalculateClick(Sender: TObject);

procedure btnCloseClick(Sender: TObject);

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

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

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

procedure FormCreate(Sender: TObject);

private

public

end;

var

Form1: TForm1;

implementation

{$R \*.lfm}

{ TForm1 }

procedure TForm1.btnCalculateClick(Sender: TObject);

const

MaxResult = 1.7e308; // Maximum representable positive finite value for a Double data type in Pascal

MinResult = -1.7e308;

var

Principal, Rate, Time, Result: Double;

begin

if (edtPrincipal.Text = '') or (edtRate.Text = '') or (edtTime.Text = '') then

begin

ShowMessage('Пожалуйста, заполните все поля.');

end

else

begin

try

Principal := StrToFloat(edtPrincipal.Text);

Rate := StrToFloat(edtRate.Text);

Time := StrToFloat(edtTime.Text);

case cbxCalculationType.ItemIndex of

0: // Простые проценты

begin

Result := Principal \* (1 + Time \* (Rate / 365));

memResult.Lines.Clear;

if (Result <= MaxResult) and (Result >= MinResult) then

memResult.Lines.Add('Результат: ' + FloatToStr(Result))

else

ShowMessage('Переполнение числа');

//memResult.Lines.Add('Результат: ' + FloatToStr(Result));

end;

1: // Сложные проценты

begin

Result := Principal \* (1 + (Rate / 100 \* Time));

memResult.Lines.Clear;

if (Result <= MaxResult) and (Result >= MinResult) then

memResult.Lines.Add('Результат: ' + FloatToStr(Result))

else

ShowMessage('Переполнение числа');

// memResult.Lines.Add('Результат: ' + FloatToStr(Result));

end;

end;

except

on E: Exception do

ShowMessage('Ошибка: ' + E.Message);

end;

end;

end;

procedure TForm1.btnCloseClick(Sender: TObject);

begin

close;

end;

procedure TForm1.edtPrincipalKeyPress(Sender: TObject; var Key: char);

begin

if not (Key in ['0'..'9', ',', '.', #8, #13, #27]) then

Key := #0;

end;

procedure TForm1.edtRateKeyPress(Sender: TObject; var Key: char);

begin

if not (Key in ['0'..'9', ',', '.', #8, #13, #27]) then

Key := #0;

end;

procedure TForm1.edtTimeKeyPress(Sender: TObject; var Key: char);

begin

if not (Key in ['0'..'9', ',', '.', #8, #13, #27]) then

Key := #0;

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

cbxCalculationType.Items.Add('Простые проценты');

cbxCalculationType.Items.Add('Сложные проценты');

cbxCalculationType.ItemIndex := 0;

end;

end.

1. **Результат выполнения программы**

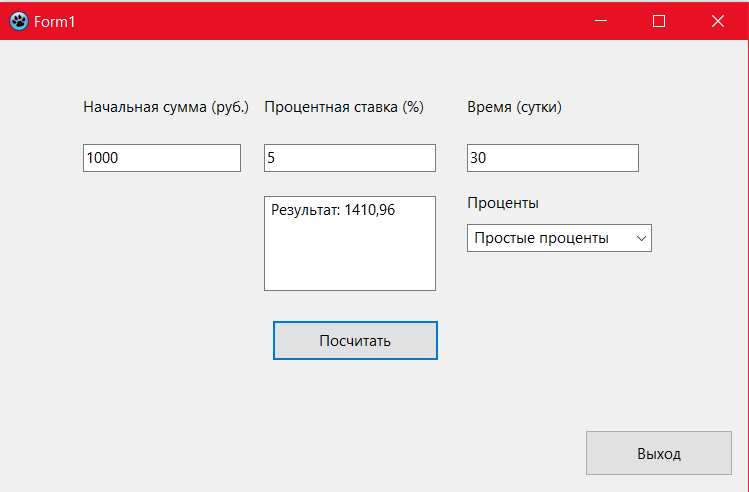


Рисунок 2 – Результат выполнения расчета с простыми процентами

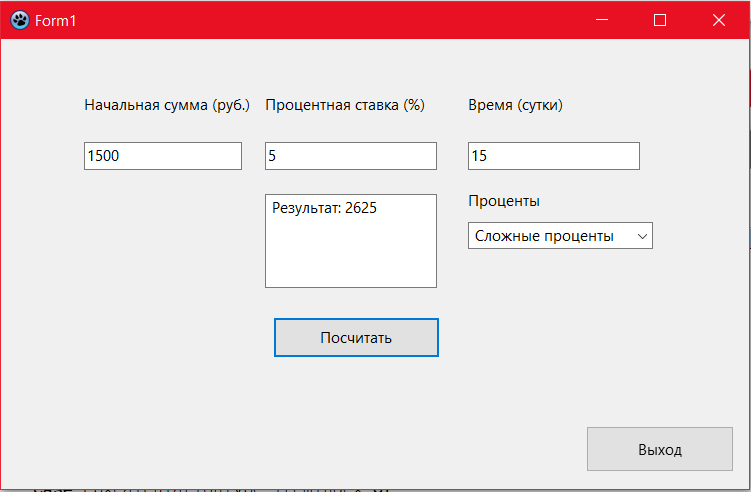


Рисунок 3 – Результат выполнения расчета с сложными процентами

1. **Вывод**

Нами была поставлена цель работы: Получение базовых навыков реализации приложений с графическим интерфейсом пользователя на основе событийно-ориентированной парадигмы. Работа была разделена на несколько этапов: цель, формулировка задания, написание кода программы, результат выполнения программы, вывод. Код программы был написан на Lazarus. Итак, делая вывод, хочу сказать, что в ходе работы все цели были достигнуты.