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

Леушина Анна Станиславовна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**2. Формулировка задания (Вариант: 15)**

1. Реализовать приложение «Калькулятор простых и сложных процентов».
2. За период начисления процентов должен приниматься месяц.
3. Предусмотреть возможность вывода пояснительной информации по расчетам.

**3. Описание алгоритма**

Простые проценты – это проценты, которые начисляются только на первоначальную сумму без учета уже начисленных процентов. Увеличение суммы происходит за счет добавления заданного процента к начальной сумме.

Сложные проценты – это проценты, которые начисляются не только на первоначальную сумму, но и на уже начисленные проценты. Увеличение суммы происходит за счет начисления процентов на сумму, которая уже увеличилась от предыдущих процентов.

**4. Код программы:**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

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

Buttons, Math;

type

{ TForm1 }

TForm1 = class(TForm)

btnClear: TButton;

btnClose: TBitBtn;

btnCalculate: TButton;

cbxCalculationType: TComboBox;

edtPrincipal: TEdit;

edtRate: TEdit;

edtTime: TEdit;

Image1: TImage;

Image2: TImage;

LabelPrincipal: TLabel;

LabelProcent: TLabel;

LabelRate: TLabel;

LabelTime: TLabel;

memResult: TMemo;

procedure btnCalculateClick(Sender: TObject);

procedure btnClearClick(Sender: TObject);

procedure btnCloseClick(Sender: TObject);

procedure cbxCalculationTypeChange(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.btnClearClick(Sender: TObject);

begin

edtPrincipal.Clear;

edtRate.Clear;

edtTime.Clear;

memResult.Clear

end;

procedure TForm1.btnCalculateClick(Sender: TObject);

const

MaxResult = 1.1e38;

MinResult = -1.1e38;

var

Principal, Rate, Time, Result: Double;

begin

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

begin

ShowMessage('Нужно заполнить 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+ (Principal \* (Rate/100/ 12) \* Time);

memResult.Lines.Clear;

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

memResult.Lines.Add('Итоговая сумма: ' + FormatFloat('0.00', Result))

else

ShowMessage('Переполнение числа');

end;

1: // Сложные проценты

begin

Result := Principal \* Power((1+((Rate/100)/12)),Time);

memResult.Lines.Clear;

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

memResult.Lines.Add('Итоговая сумма: ' + FormatFloat('0.00', Result))

else

ShowMessage('Переполнение числа');

end;

end;

except

on E: Exception do

ShowMessage('Ошибка: ' + E.Message);

end;

end;

end;

procedure TForm1.btnCloseClick(Sender: TObject);

begin

close;

end;

procedure TForm1.cbxCalculationTypeChange(Sender: TObject);

begin

case cbxCalculationType.ItemIndex of

0: begin

Image1.Visible := True;

Image2.Visible := False;

end;

1: begin

Image1.Visible := False;

Image2.Visible := True;

end;

end;

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;

cbxCalculationType.OnChange := @cbxCalculationTypeChange;

Image1.Picture.LoadFromFile('1.jpg');

Image2.Picture.LoadFromFile('2.jpg');

Image1.Visible := True;

Image2.Visible := False;

end;

end.

**5. Результат выполнения программы**

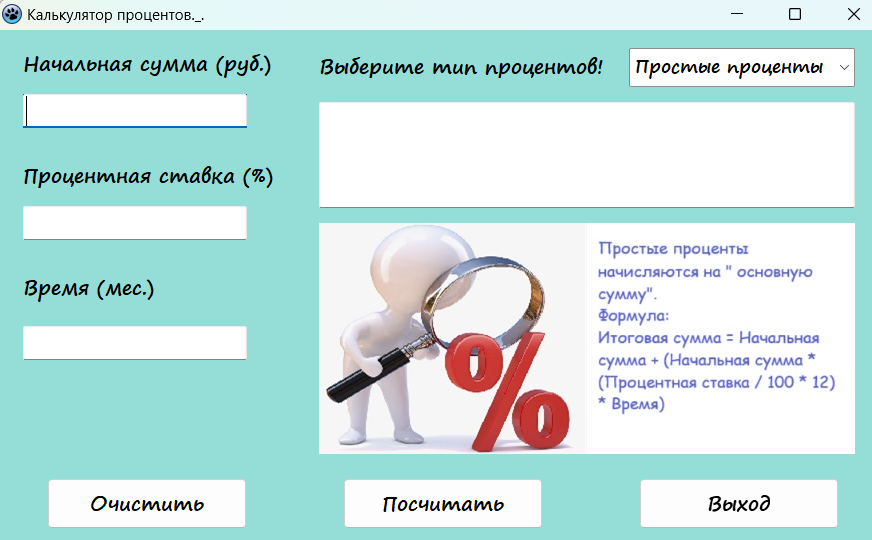


Рисунок 1 – Окно для расчета простых процентов

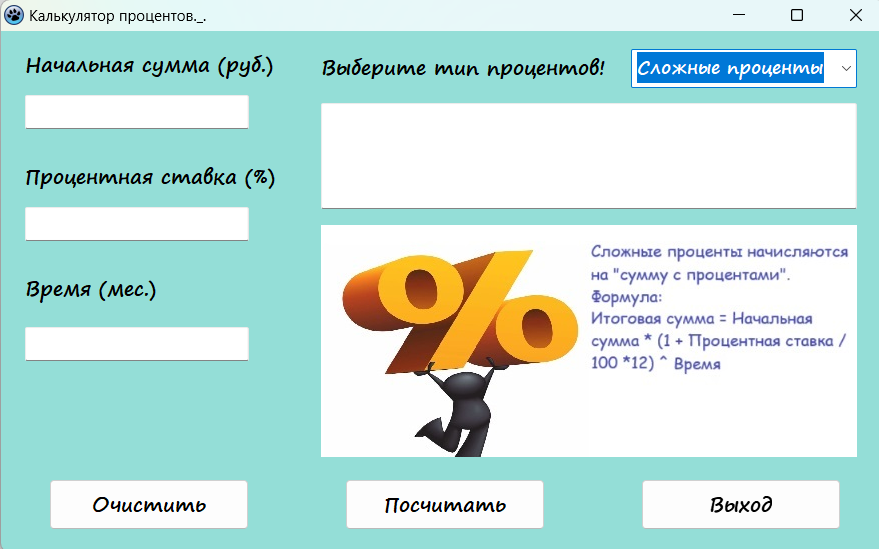


Рисунок 2 – Окно для расчета сложных процентов

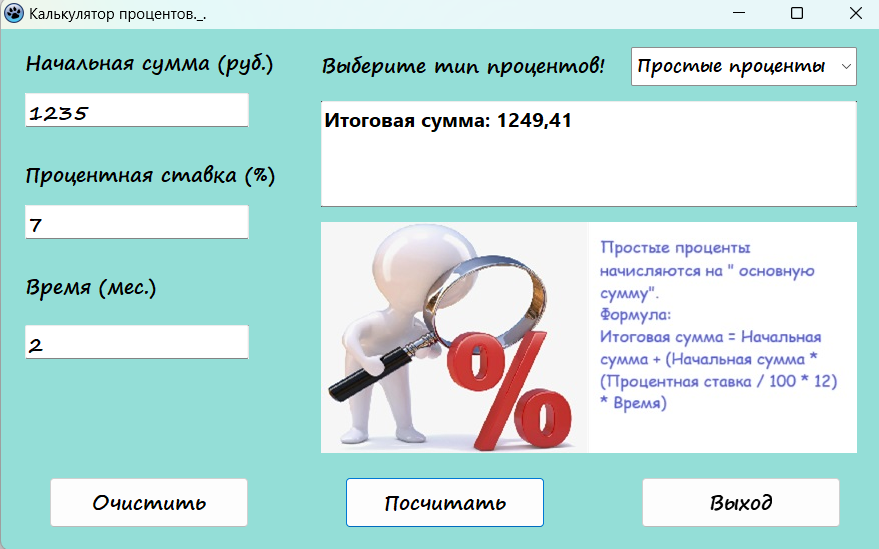


Рисунок 3 – Пример расчета простых процентов

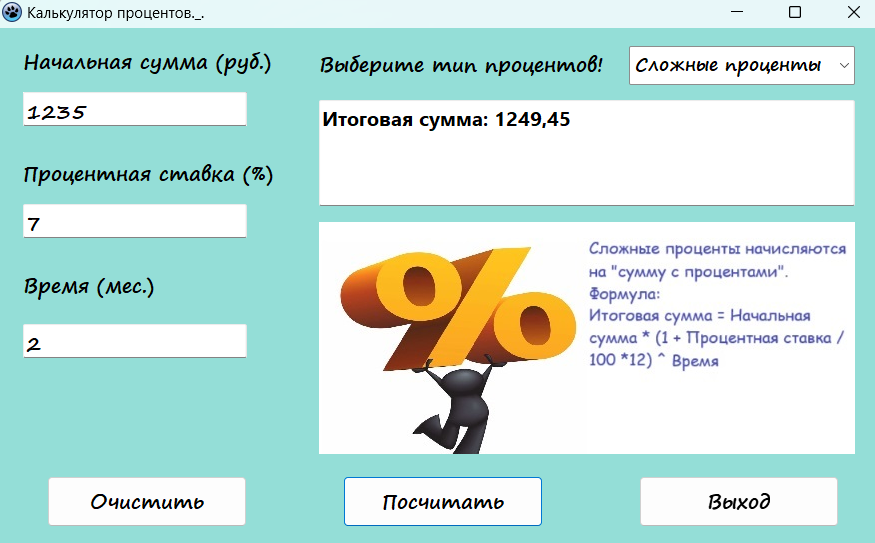


Рисунок 4 – Пример расчета сложных процентов

**6. Вывод**

В домашней контрольной работе №8 была реализована программа для расчета итоговой суммы с учетом простых и сложных процентов в среде разработки Lazarus. Были определены необходимые формулы для расчета процентов, затем был разработан пользовательский интерфейс и реализована логика программы. В ходе выполнения работы затруднений не возникло, все поставленные задачи были выполнены в полном объеме.