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-202-52-00

Савватеев Аркадий Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

**Цель работы**

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

**Формулировка задания (с вариантом)**

Вариант 13

Задание:

1. Реализовать приложение "Калькулятор простых и сложных процентов".

2. За период начисления процентов должен приниматься день.

3. Предусмотреть возможность вывода пояснительной информации по расчетам.

**Описание приложения**

Калькулятор считает размер полученных денег за выбранный пользователем период времени, процент вклада и размер суммы.

Окно приложения содержит 5 типов элементов:

Элемент TForm — основное окно программы;

Элемент TLabel — текстовая надпись;

Элемент TEdit — строка ввода;

Элемент TButton — кнопки выполнения действий;

Элемент TBitBtn — кнопка закрытия окна.

Элемент TComboBox — кнопка с выпадающим списком.

**Код программы**

**unit Unit1;**

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

**interface**

**uses**

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

**Buttons, Math;**

**type**

**{ TForm1 }**

**TForm1 = class(TForm)**

**BitBtn1: TBitBtn;**

**Button1: TButton;**

**InterestType: TComboBox;**

**Time: TComboBox;**

**Label5: TLabel;**

**Sum: TEdit;**

**Percent: TEdit;**

**Period: TEdit;**

**Label1: TLabel;**

**Label2: TLabel;**

**Label3: TLabel;**

**Label4: TLabel;**

**Output: TMemo;**

**procedure BitBtn1Click(Sender: TObject);**

**procedure Button1Click(Sender: TObject);**

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

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

**procedure Label2Click(Sender: TObject);**

**procedure Label4Click(Sender: TObject);**

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

**private**

**public**

**end;**

**var**

**Form1: TForm1;**

**implementation**

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

**{ TForm1 }**

**procedure TForm1.InterestTypeChange(Sender: TObject);**

**begin**

**end;**

**procedure TForm1.Button1Click(Sender: TObject);**

**var**

**x, a, b, f: Float;**

**i: set of char;**

**j: Integer;**

**k: Integer;**

**Valid: Boolean;**

**begin**

**i := ['0'..'9', ','];**

**j := 1;**

**if (Period.Text = '') or (InterestType.Text = '') or (Sum.Text = '') or (Percent.Text = '') then**

**begin**

**ShowMessage('Не все данные введены!');**

**Exit;**

**end;**

**Valid := True;**

**for k := 1 to Length(Period.Text) do**

**if not (Period.Text[k] in i) then**

**Valid := False;**

**if Valid then**

**for k := 1 to Length(Sum.Text) do**

**if not (Sum.Text[k] in i) then**

**Valid := False;**

**if Valid then**

**for k := 1 to Length(Percent.Text) do**

**if not (Percent.Text[k] in i) then**

**Valid := False;**

**if not Valid then**

**begin**

**ShowMessage('Некорректно введены данные! Используйте только цифры и запятую.');**

**Exit;**

**end;**

**if InterestType.Text = 'Простые проценты' then**

**begin**

**if Time.Text = 'Дни' then**

**begin**

**a := StrToFloat(Sum.Text);**

**b := StrToFloat(Percent.Text) / 100;**

**x := StrToFloat(Period.Text);**

**f := a \* b \* (x/365);**

**Output.Lines.Add('Будет начислено с суммы в ' + FloatToStr(a) + ' рублей с процентной ставкой в ' + FloatToStr(b) + ' начисляемых ежедневно: ' + FloatToStr(f)+ 'рублей');**

**end;**

**if Time.Text = 'Недели' then**

**begin**

**a := StrToFloat(Sum.Text);**

**b := StrToFloat(Percent.Text) / 100;**

**x := StrToFloat(Period.Text);**

**f := a \* b \* (x/52);**

**Output.Lines.Add('Будет начислено с суммы в ' + FloatToStr(a) + ' рублей с процентной ставкой в ' + FloatToStr(b) + ' начисляемых ежедневно: ' + FloatToStr(f)+ 'рублей');**

**end;**

**if Time.Text = 'Месяц' then**

**begin**

**a := StrToFloat(Sum.Text);**

**b := StrToFloat(Percent.Text) / 100;**

**x := StrToFloat(Period.Text);**

**f := a \* b \* (x/12);**

**Output.Lines.Add('Будет начислено с суммы в ' + FloatToStr(a) + ' рублей с процентной ставкой в ' + FloatToStr(b) + ' начисляемых ежедневно: ' + FloatToStr(f)+ 'рублей');**

**end;**

**if Time.Text = 'Квартал' then**

**begin**

**a := StrToFloat(Sum.Text);**

**b := StrToFloat(Percent.Text) / 100;**

**x := StrToFloat(Period.Text);**

**f := a \* b \* (x/4);**

**Output.Lines.Add('Будет начислено с суммы в ' + FloatToStr(a) + ' рублей с процентной ставкой в ' + FloatToStr(b) + ' начисляемых ежедневно: ' + FloatToStr(f)+ 'рублей');**

**end;**

**if Time.Text = 'Год' then**

**begin**

**a := StrToFloat(Sum.Text);**

**b := StrToFloat(Percent.Text) / 100;**

**x := StrToFloat(Period.Text);**

**f := a \* b \* x;**

**Output.Lines.Add('Будет начислено с суммы в ' + FloatToStr(a) + ' рублей с процентной ставкой в ' + FloatToStr(b) + ' начисляемых ежедневно: ' + FloatToStr(f)+ 'рублей');**

**end;**

**end;**

**if InterestType.Text = 'Сложные проценты' then**

**begin**

**if Time.Text = 'Дни' then**

**begin**

**a := StrToFloat(Sum.Text);**

**b := StrToFloat(Percent.Text) / 100;**

**x := StrToFloat(Period.Text);**

**f := a \* b \* (x/365);**

**Output.Lines.Add('Будет начислено с суммы в ' + FloatToStr(a) + ' рублей с процентной ставкой в ' + FloatToStr(b) + ' начисляемых ежедневно: ' + FloatToStr(f)+ 'рублей');**

**end;**

**if Time.Text = 'Недели' then**

**begin**

**a := StrToFloat(Sum.Text);**

**b := StrToFloat(Percent.Text) / 100;**

**x := StrToFloat(Period.Text);**

**f := a \* b \* (x/52);**

**Output.Lines.Add('Будет начислено с суммы в ' + FloatToStr(a) + ' рублей с процентной ставкой в ' + FloatToStr(b) + ' начисляемых ежедневно: ' + FloatToStr(f)+ 'рублей');**

**end;**

**if Time.Text = 'Месяц' then**

**begin**

**a := StrToFloat(Sum.Text);**

**b := StrToFloat(Percent.Text) / 100;**

**x := StrToFloat(Period.Text);**

**f := a \* b \* (x/12);**

**Output.Lines.Add('Будет начислено с суммы в ' + FloatToStr(a) + ' рублей с процентной ставкой в ' + FloatToStr(b) + ' начисляемых ежедневно: ' + FloatToStr(f)+ 'рублей');**

**end;**

**if Time.Text = 'Квартал' then**

**begin**

**a := StrToFloat(Sum.Text);**

**b := StrToFloat(Percent.Text) / 100;**

**x := StrToFloat(Period.Text);**

**f := a \* b \* (x/4);**

**Output.Lines.Add('Будет начислено с суммы в ' + FloatToStr(a) + ' рублей с процентной ставкой в ' + FloatToStr(b) + ' начисляемых ежедневно: ' + FloatToStr(f)+ 'рублей');**

**end;**

**if Time.Text = 'Год' then**

**begin**

**a := StrToFloat(Sum.Text);**

**b := StrToFloat(Percent.Text) / 100;**

**x := StrToFloat(Period.Text);**

**f := a \* b \* x;**

**Output.Lines.Add('Будет начислено с суммы в ' + FloatToStr(a) + ' рублей с процентной ставкой в ' + FloatToStr(b) + ' начисляемых ежедневно: ' + FloatToStr(f)+ 'рублей');**

**end;**

**end;**

**if InterestType.Text = 'Сложные проценты' then**

**begin**

**a := StrToFloat(Sum.Text);**

**b := StrToFloat(Percent.Text) / 100;**

**x := StrToFloat(Period.Text);**

**if Time.Text = 'Дни' then**

**begin**

**f := a \* Power(1 + b/365, x);**

**Output.Lines.Add('Итоговая сумма за ' + FloatToStr(x) + ' дней: ' + FloatToStrF(f, ffCurrency, 12, 2));**

**end**

**else if Time.Text = 'Недели' then**

**begin**

**f := a \* Power(1 + b/52, x);**

**Output.Lines.Add('Итоговая сумма за ' + FloatToStr(x) + ' недель: ' + FloatToStrF(f, ffCurrency, 12, 2));**

**end**

**else if Time.Text = 'Месяц' then**

**begin**

**f := a \* Power(1 + b/12, x);**

**Output.Lines.Add('Итоговая сумма за ' + FloatToStr(x) + ' месяцев: ' + FloatToStrF(f, ffCurrency, 12, 2));**

**end**

**else if Time.Text = 'Квартал' then**

**begin**

**f := a \* Power(1 + b/4, x);**

**Output.Lines.Add('Итоговая сумма за ' + FloatToStr(x) + ' кварталов: ' + FloatToStrF(f, ffCurrency, 12, 2));**

**end**

**else if Time.Text = 'Год' then**

**begin**

**f := a \* Power(1 + b, x);**

**Output.Lines.Add('Итоговая сумма за ' + FloatToStr(x) + ' лет: ' + FloatToStrF(f, ffCurrency, 12, 2));**

**end;**

**end;**

**end;**

**procedure TForm1.BitBtn1Click(Sender: TObject);**

**begin**

**Close;**

**end;**

**procedure TForm1.FormCreate(Sender: TObject);**

**begin**

**end;**

**procedure TForm1.Label2Click(Sender: TObject);**

**begin**

**end;**

**procedure TForm1.Label4Click(Sender: TObject);**

**begin**

**end;**

**procedure TForm1.SumChange(Sender: TObject);**

**begin**

**end;**

**end.**

**Результат выполнения программы**

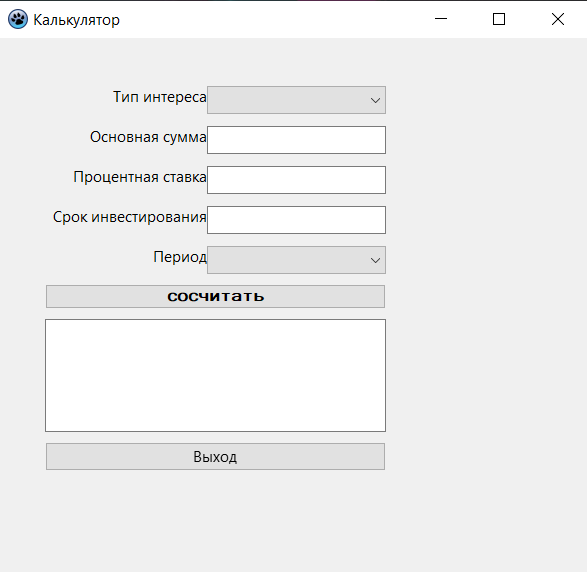


Рисунок 1 — Результат выполнения программы

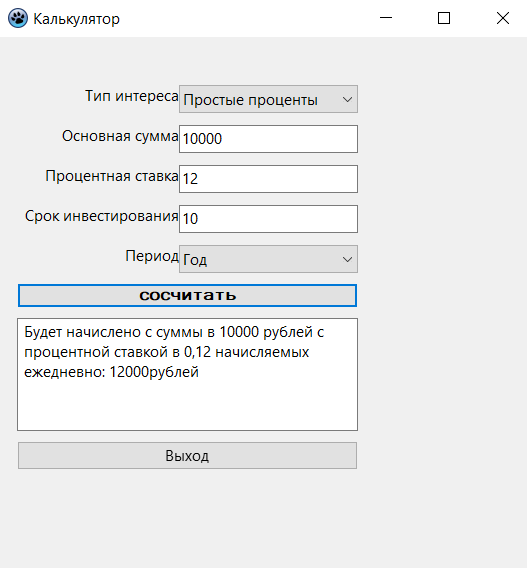


Рисунок 2 – Пример работы для подсчета

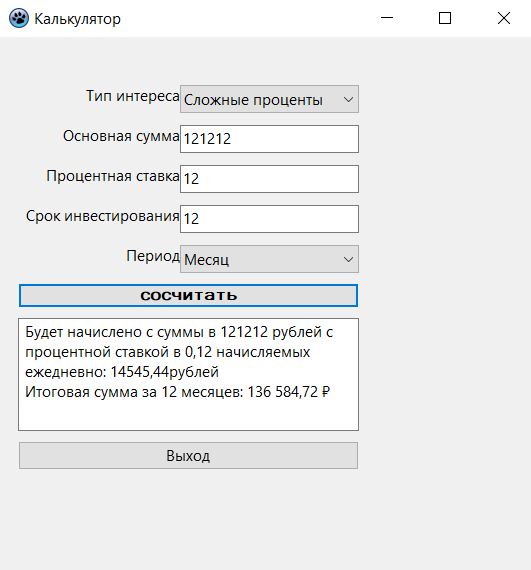


Рисунок 3 – подсчет сложных процентов

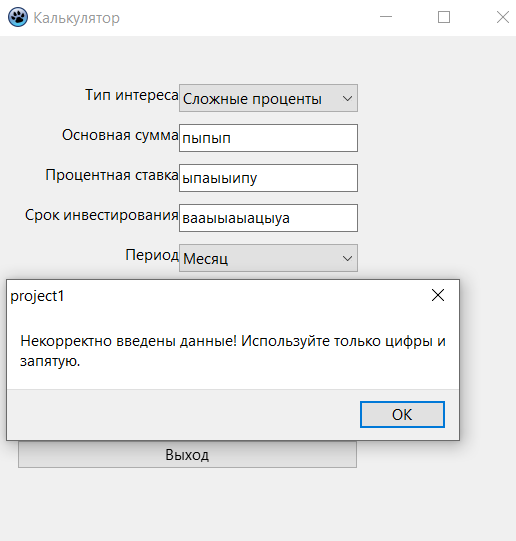


Рисунок 4 – пример вывода об ошибке

**Вывод**

В процессе работы над проектом создания калькулятора целочисленной арифметики в среде разработки Lazarus я углубился в основы событийно ориентированного программирования.

Ключевыми аспектами моей работы стали:

1. Структура проекта: Я организовал код в соответствии с принципами событийно ориентированного подхода, определив отдельные обработчики для каждого события, такие как нажатие кнопок и изменение значений.

2. Интерфейс пользователя: Создание удобного и интуитивно понятного интерфейса позволило пользователю легко взаимодействовать с калькулятором. Я использовал визуальные компоненты Lazarus, такие как кнопки и текстовые поля.

3. Обработка ошибок: Я разработал механизм для обработки ошибок, таких как деление на ноль, что значительно улучшило пользовательский опыт и надежность программы.

В результате, благодаря практическому применению основ событийно ориентированного программирования, мне удалось создать функциональный и удобный в использовании калькулятор. Этот проект не только закрепил мои знания в области программирования, но и дал ценный опыт работы с визуальными компонентами и обработкой событий в Lazarus.