Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №3**

**«Основы событийно-ориентированного программирования»**

**ПО МДК 05.02 «Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-202-52-00

Ситников Иван Владимирович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**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 алгоритма.**

Вкладывая деньги в банк, мы хотим не только сохранить существующие средства, но и получить доход – проценты. В таком случае мы стремимся выбрать вклад с максимальной доходностью.

Существуют два вида процентов по вкладам: простые и сложные. Сложные проценты отличаются от простых наличием капитализации.

В данном задании предполагается что пользователь указывает количество дней вклада, а значит капитализация для сложных процентов будет ежедневной.

Простые проценты вычисляются по формуле:

S = (P \* I \* t / 365) / 100

Обозначения:

S – прибыль со вклада (только проценты, без тела вклада);

P – сумма, изначально внесенная на депозит;

I – размер % ставки (за год);

t – кол-во дней начисления %;

Сложные проценты с ежедневной капитализацией рассчитываются по формуле:

S = P \* (1 + (N/100)/365)^T, где:

S – суммарный доход (тело вклада + проценты);

Р – внесенная при заключении договора сумма;

N – годовая % ставка;

Т – кол-во дней, на которые открыт депозит.

**Схема алгоритма**

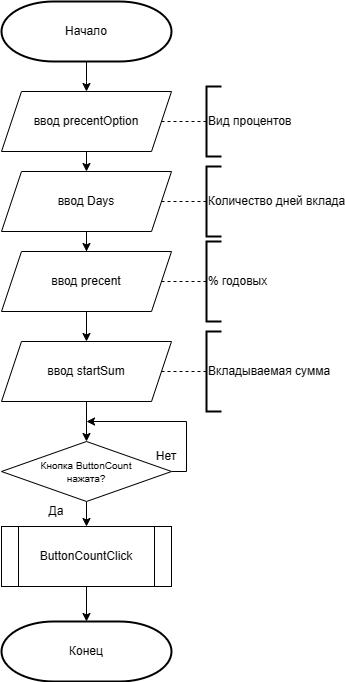


Рисунок 1 – Основное тело программы

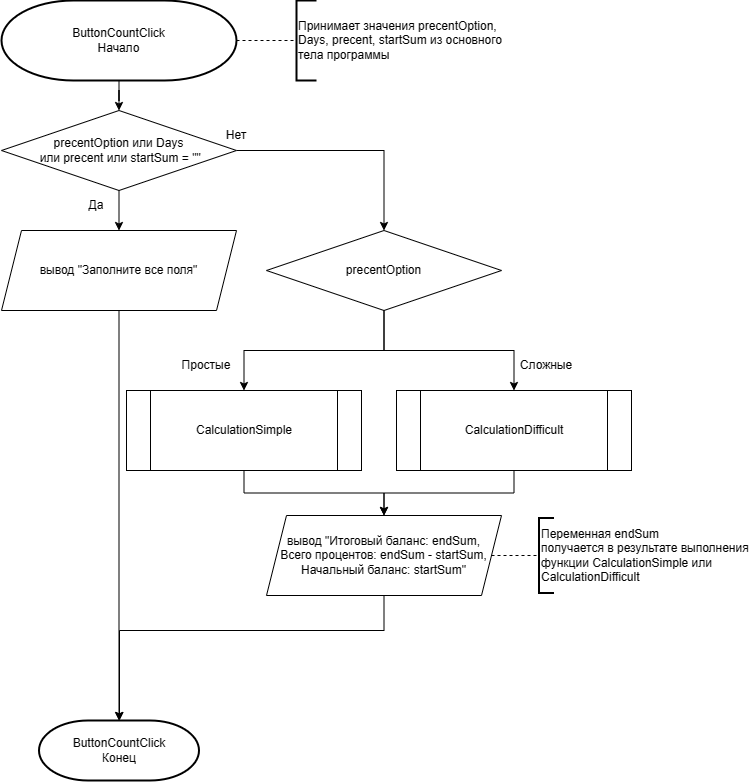


Рисунок 2 – Процедура, обработки нажатия на кнопку «Считать»



Рисунок 3 – Функция расчета простых процентов

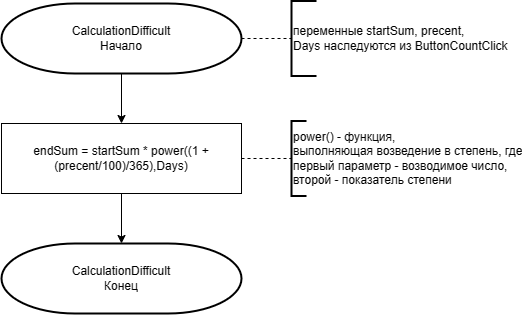


Рисунок 4 – Функция расчета сложных процентов

**Результат выполнения программы**

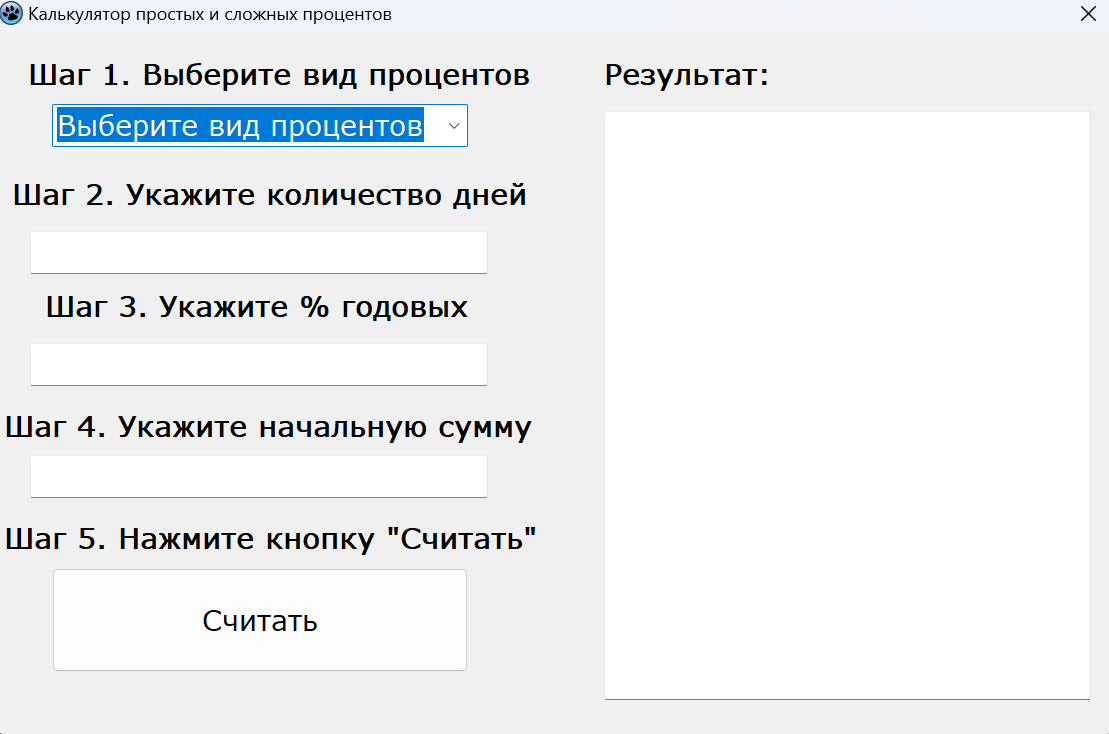
****

Рисунок 5 – Окно программы

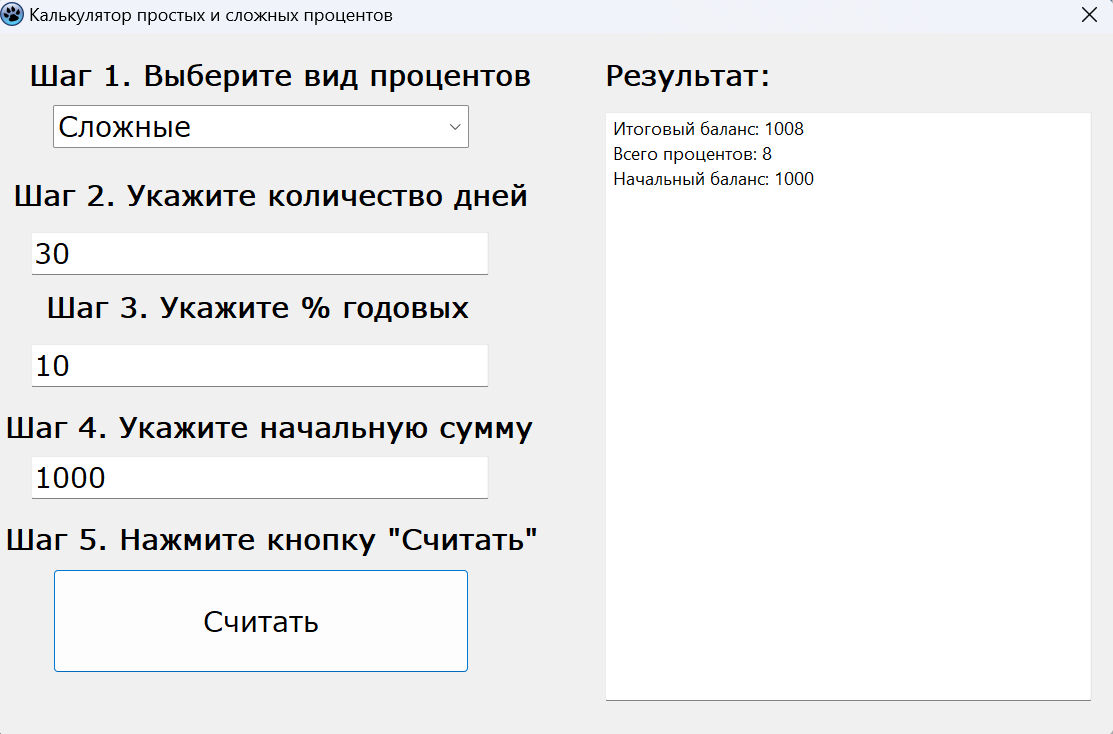
****

Рисунок 6 – Окно программы с расчетами

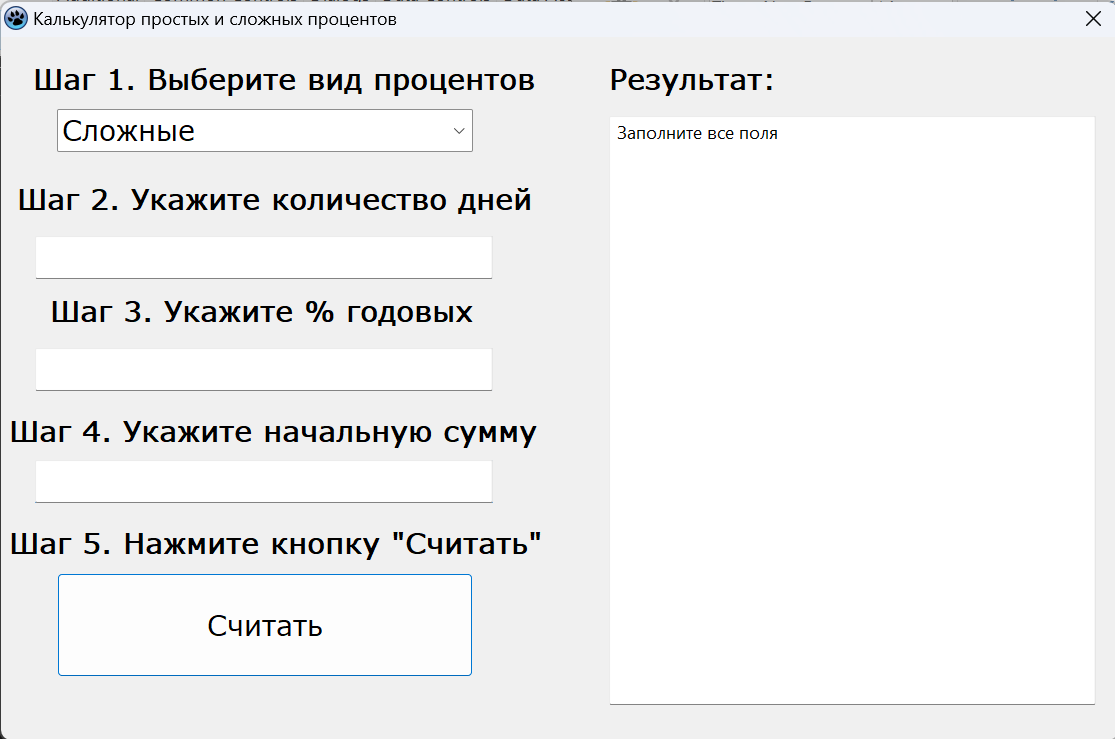


Рисунок 7 – Окно программы с обработкой ошибки

**Код программы**

unit Main;

{$mode objfpc}{$H+}

interface

uses

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

type

{ TForm1 }

TForm1 = class(TForm)

ButtonCount: TButton;

ComboBoxChoiceOption: TComboBox;

EditQuantityDays: TEdit;

EditPercent: TEdit;

EditStartValue: TEdit;

LabelChoiceOption: TLabel;

LabelQuantityDays: TLabel;

LabelPercent: TLabel;

LabelResult: TLabel;

LabelStartValue: TLabel;

LabelStartValue1: TLabel;

MemoResult: TMemo;

procedure ButtonCountClick(Sender: TObject);

procedure ComboBoxChoiceOptionChange(Sender: TObject);

procedure EditPercentChange(Sender: TObject);

procedure EditQuantityDaysChange(Sender: TObject);

procedure EditStartValueChange(Sender: TObject);

private

public

end;

var

Form1: TForm1;

precentOption: String;

Days: Integer;

Precent, startSum, res, endSum: Double;

implementation

{$R \*.lfm}

{ TForm1 }

//Вид процентов

procedure TForm1.ComboBoxChoiceOptionChange(Sender: TObject);

begin

precentOption := ComboBoxChoiceOption.Text;

end;

//Рассчет сложных процентов

function CalculationDifficult(startSum, Precent: Double; Days: Integer):Double;

begin

res:= startSum \* power((1 + (Precent/100)/365),Days);

CalculationDifficult := res;

end;

//Рассчет простых процентов

function CalculationSimple(startSum, Precent: Double; Days: Integer):Double;

begin

res:= (startSum \* Precent \* Days / 365) / 100;

CalculationSimple := res;

end;

//Начать рассчеты

procedure TForm1.ButtonCountClick(Sender: TObject);

begin

if (EditPercent.Text = '')and(EditQuantityDays.Text='')and(EditStartValue.Text='')

then begin

MemoResult.Text := 'Заполните все поля';

exit;

end;

//Обработка исключения, если пользователь введёт неверные данные

try

Precent := StrToFloat(EditPercent.Text);

Days := StrToInt(EditQuantityDays.Text);

startSum := StrToFloat(EditStartValue.Text);

except begin

MemoResult.Text := 'Некорректный ввод';

exit;

end;

end;

if (precentOption = 'Простые') then

endSum := CalculationSimple(startSum, Precent, Days)+startSum

else if (precentOption = 'Сложные') then

endSum := CalculationDifficult(startSum, Precent, Days)

else

begin

MemoResult.Text := 'Ошибка подсчета';

exit;

end;

MemoResult.Text := 'Итоговый баланс: '+FloatToStr(round(endSum))+LineEnding

+'Всего процентов: '+ FloatToStr(round(endSum-startSum))+LineEnding

+'Начальный баланс: '+ FloatToStr(startSum);

end;

//% годовых

procedure TForm1.EditPercentChange(Sender: TObject);

begin

if EditPercent.Text = '' then exit;

end;

//Количество дней

procedure TForm1.EditQuantityDaysChange(Sender: TObject);

begin

if EditQuantityDays.Text = '' then exit;

end;

// Начальная сумма

procedure TForm1.EditStartValueChange(Sender: TObject);

begin

if EditStartValue.Text = '' then exit;

end;

end.

**Вывод**

В ходе написания программы «Калькулятор простых и сложных процентов» были закреплены знания об объектно-событийной парадигме программирования, а также знания об обработке событий в среде разработки Lazarus.

В конечном итоге был разработан «Калькулятор простых и сложных процентов», осуществляющий расчет прибыли со вклада, при вводимых пользователем условиях: вид процентов, количество дней вклада, количество % годовых и начальную сумму.