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

Конышев Иван Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров 2024

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

**Задание (Вариант 7)**

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**

1. Проверяем значение ComboBox1.Text. Если оно равно 'Вид процента', выводим сообщение 'Выберите вид процента'.

2. Проверяем, чтобы поля EditSum, EditRate и EditDays были заполнены. Если хотя бы одно из полей пустое, выводим сообщение об ошибке ввода.

3. Если все поля заполнены, преобразуем текст из полей в числовые значения: Sum, Rate и Days.

4. Вычисляем простой и сложный процент по формулам SimpleInterest и CompoundInterest.

5. Если ComboBox1.Text равно 'Простой процент', очищаем MemoResult и вычисляем итоговую сумму с простым процентом.

6. Выводим результат простого процента и итоговую сумму в MemoResult.

7. Если ComboBox1.Text не равно 'Простой процент', очищаем MemoResult и вычисляем итоговую сумму с сложным процентом.

8. Выводим результат сложного процента и итоговую сумму в MemoResult.

9. Кнопка "Button1" (по умолчанию выход) - при нажатии на нее происходит закрытие окна или формы, в которой находится данная кнопка.

10. Кнопка "ButtonClear" - при нажатии на нее происходит очистка содержимого следующих элементов управления:

- MemoResult: это многострочное поле для вывода информации или результата;

- EditSum: это поле для ввода суммы;

- EditRate: это поле для ввода процентной ставки;

- EditDays: это поле для ввода количества дней.

**Код основной программы**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

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

Math;

type

{ TForm1 }

TForm1 = class(TForm)

Button1: TButton;

ButtonCalculate: TButton;

ButtonClear: TButton;

ComboBox1: TComboBox;

EditSum: TEdit;

EditRate: TEdit;

EditDays: TEdit;

Image1: TImage;

LabelRate: TLabel;

LabelSum: TLabel;

LabelDays: TLabel;

MemoResult: TMemo;

procedure Button1Click(Sender: TObject);

procedure ButtonCalculateClick(Sender: TObject);

procedure ButtonClearClick(Sender: TObject);

procedure MemoResultChange(Sender: TObject);

private

public

end;

var

Form1: TForm1;

implementation

{$R \*.lfm}

{ TForm1 }

procedure TForm1.ButtonCalculateClick(Sender: TObject);

var

Sum, Rate, Days, SimpleInterest, CompoundInterest, ItogResult: Double;

begin

if ComboBox1.Text= 'Вид процента' then

Showmessage('Выберите вид процента');

if (EditSum.Text='') or (EditRate.Text='') or (EditDays.Text='') then

Showmessage('Ошибка ввода')

else

begin

Sum := StrToFloat(EditSum.Text);

Rate := StrToFloat(EditRate.Text) / 100;

Days := StrToInt(EditDays.Text);

SimpleInterest := Sum \* Rate \* Days;

CompoundInterest := Sum \* Power((1 + Rate), Days) - Sum;

if ComboBox1.Text= 'Простой процент' then

begin

MemoResult.Clear;

ItogResult := Sum\*(1+SimpleInterest);

MemoResult.Lines.add('Процент = ' + Format('%f', [SimpleInterest]));

MemoResult.Lines.add('Итоговая сумма = ' + Format('%f', [ItogResult]));

end

else

MemoResult.Clear;

ItogResult := Sum\*(1+CompoundInterest);

MemoResult.Lines.Add('Процент: ' + Format('%f', [CompoundInterest]));

MemoResult.Lines.add('Итоговая сумма = ' + Format('%f', [ItogResult]));

end;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

close;

end;

procedure TForm1.ButtonClearClick(Sender: TObject);

begin

MemoResult.Clear;

EditSum.Clear;

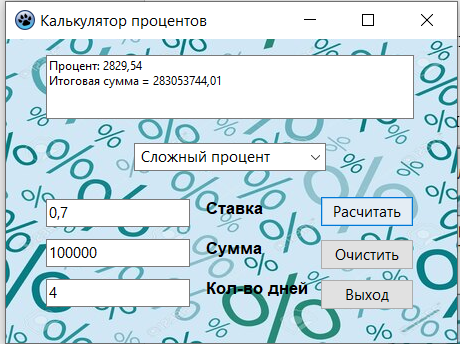
EditRate.Clear;

EditDays.Clear;

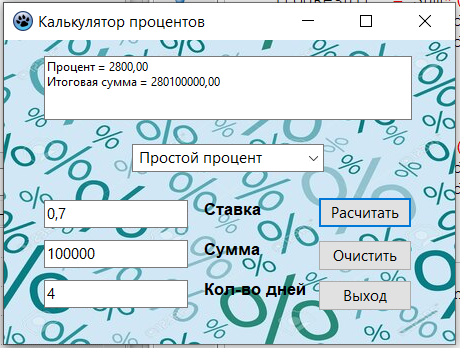
end;

**Результат выполнения программы**

1. Сложный процент



1. Простой процент



Вывод: путем изучения и практического применения алгоритмов обработки событий интерфейса, разработчики смогут понять принципы взаимодействия пользователя с приложением, а также научиться эффективно управлять поведением элементов управления в соответствии с действиями пользователя.

Осуществление поставленной цели позволит развить не только технические навыки программирования, но и способствует лучшему пониманию процессов интерактивной обратной связи между пользователем и программой. Полученные навыки могут быть полезны как для создания игровых приложений, так и для разработки более сложных программ с интерфейсом пользователя. В целом, достижение поставленной цели позволит разработчикам стать более компетентными в области создания приложений с графическим интерфейсом на основе событийно-ориентированной парадигмы.