Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №1**

**«ОСНОВЫ СОБЫТИЙНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИДМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-204-52-00

Утёмова Ксения Александровна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

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

**Формулировка задания:**

Изображение выглядит как текст, снимок экрана, компьютер, программное обеспечение

Автоматически созданное описание

Рисунок 1 – Формулировка задания

**Описание алгоритма:**

С помощью кнопок (Запустить, остановить и сброс) можно управлять таймером, который связан с отображением счетчика секундомера на экране.

Кнопка «Запустить» включает таймер, и активирует кнопки «Остановить» и «Сброс» для взаимодействия с ними.

Кнопка «Остановить» останавливает таймер и вписывает его результат в область для отображения значений секундомера при остановке.

Кнопка «Сброс» останавливает работу таймера, обнуляет изображение счетчика и очищает область для данных со значениями остановки секундомера.

Кнопка «Выход» выходит из приложения.

**Код программы:**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

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

type

{ TSecond }

TSecond = class(TForm)

btnStart: TButton;

btnStop: TButton;

btnReset: TButton;

Button1: TButton;

Image1: TImage;

Label1: TLabel;

lblTimer: TLabel;

memoLaps: TMemo;

TimerTime: TTimer;

procedure btnResetClick(Sender: TObject);

procedure btnStartClick(Sender: TObject);

procedure btnStopClick(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure TimerTimeTimer(Sender: TObject);

private

{ private declarations }

FStartTime: TDateTime;

FLapStartTime: TDateTime;

FTotalElapsed: TDateTime;

elapsedTime: string;

public

{ public declarations }

end;

var

Second: TSecond;

implementation

{$R \*.lfm}

{ TSecond }

procedure TSecond.Button1Click(Sender: TObject);

begin

Close;

end;

procedure TSecond.btnStopClick(Sender: TObject);

begin

TimerTime.Enabled := False;

btnStart.Enabled := True;

btnStop.Enabled := False;

memoLaps.Lines.Add(lblTimer.Caption);

end;

procedure TSecond.btnResetClick(Sender: TObject);

begin

TimerTime.Enabled := False;

FStartTime := 0;

FLapStartTime := 0;

FTotalElapsed := 0;

lblTimer.Caption := '00:00:00:000';

memoLaps.Clear;

btnStart.Enabled := True;

btnStop.Enabled := False;

btnReset.Enabled := False;

btnStart.Caption:='Начать';

end;

procedure TSecond.TimerTimeTimer(Sender: TObject);

var

elapsed: TDateTime;

lapElapsed: TDateTime;

begin

if TimerTime.Enabled then

begin

elapsed := Now - FStartTime;

lapElapsed := Now - FLapStartTime;

FTotalElapsed := FTotalElapsed + lapElapsed;

lblTimer.Caption := FormatDateTime('hh:nn:ss:zzz', FTotalElapsed);

FLapStartTime := Now;

end;

end;

procedure TSecond.btnStartClick(Sender: TObject);

begin

FStartTime := Now;

FLapStartTime := FStartTime;

TimerTime.Enabled := True;

btnStart.Enabled := False;

btnStop.Enabled := True;

btnReset.Enabled := True;

btnStart.Caption:='Продолжить';

end;

end.

**Результат выполнения программы:**

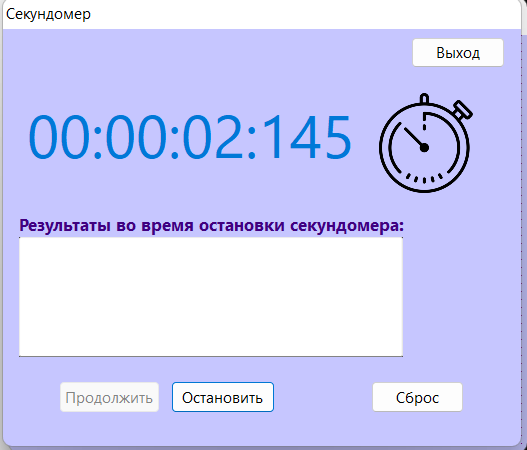


Рисунок 2 – Результат запуска секундомера

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 3. Результат остановки таймера

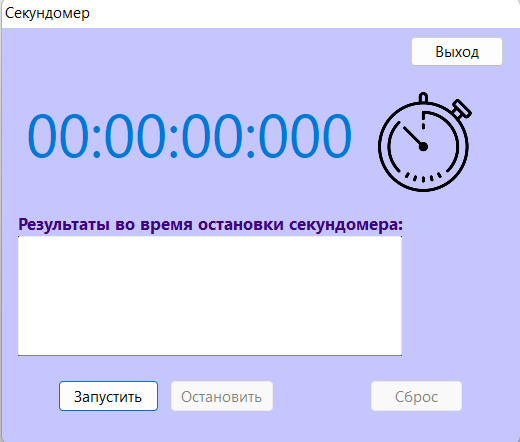


Рисунок 4 - Результат сброса секундомера