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

Коргина Екатерина Кириловна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. Цель контрольной работы.

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

1. Формулировка задания

I) Реализовать приложение «Секундомер».

II) Предусмотреть возможности временной остановки отсчета и полного сброса.

1. Описание алгоритма

При написании код программы было включено 5 основных компонентов.

1. TButton - является компонентом, представляющим собой базовый элемент управления в виде кнопки.

2. TLabel. Компонент TLabel (Надпись) используется для отображения текста в форме, который нельзя изменять непосредственно через графический интерфейс пользователя.

3. TStatic Text - оконный компонент, отображающий текст на форме, без возможности редактирования пользователем.

4. TTimer - является компонентом, расположенным на вкладке System палитры компонентов и представляющим собой таймер с интервалом в миллисекундах.

5. TBitBtn - это компонент, созданный на основе кнопки с изображением на её поверхности. TBitBtn является потомком TWinControl и доступен на вкладке Additional палитры компонентов.

1. Код программы

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

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

Buttons;

type

{ TStopWatch }

TStopWatch = class(TForm)

BitBtnExit: TBitBtn;

ButtonStart: TButton;

ButtonStop: TButton;

ButtonContinue: TButton;

Time: TLabel;

Dio: TStaticText;

Timer: TTimer;

procedure ButtonStartClick(Sender: TObject);

procedure ButtonStopClick(Sender: TObject);

procedure ButtonContinueClick(Sender: TObject);

procedure TimerTimer(Sender: TObject);

private

public

end;

var

StopWatch: TStopWatch;

Start : Double;

b: Boolean;

ZaWarka: TDateTime;

implementation

{$R \*.lfm}

function GetTime: TDateTime;

var s : TSystemTime;

begin

GetLocalTime(s);

Result := SystemTimeToDateTime(s);

end;

procedure TStopWatch.ButtonStartClick(Sender: TObject);

begin

b:= False;

Dio.Caption:='';

Timer.Enabled := True;

ZaWarka:=0;

ButtonContinue.Visible:=False;

end;

procedure TStopWatch.ButtonStopClick(Sender: TObject);

begin

Timer.Enabled := False;

Dio.Caption:='Время остановило движение';

ZaWarka:=GetTime-Start+ZaWarka;

ButtonContinue.Visible:=True;

end;

procedure TStopWatch.ButtonContinueClick(Sender: TObject);

begin

b:= False;

Dio.Caption:='';

Timer.Enabled := True;

ButtonContinue.Visible:=False;

end;

procedure TStopWatch.TimerTimer(Sender: TObject);

begin

if not (b) then

Start := GetTime;

b:= True;

Time.Caption := FormatDateTime('hh:nn:ss zzz', GetTime - Start + ZaWarka)

end;

end.

1. Результат выполнения программы

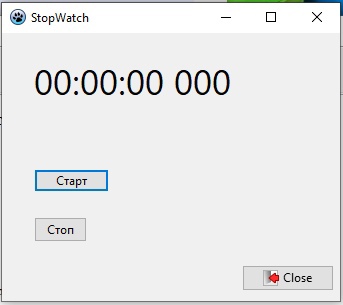


Рисунок 1 – Результат выполнения программы

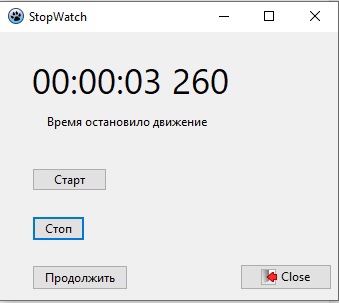


Рисунок 2 – Результат выполнения работы

1. Вывод

В ходе выполнения домашней контрольной работы №8 «Основы событийно-ориентированного программирования», были получены базовые навыки реализации приложений с графическим интерфейсом пользователя на основе событийно-ориентированного парадигмы. Приложение было написано на языке программирования Lazarus. При разработке были вставлены 5 компонентов для основной работы приложения: TBitBtn, TTimer, TStatic Text, TLabel, TButton. Так же было выполнение масштабирование приложения для улучшенного восприятия и раскрытие окна. Проблем особых не возникало, так как работа на языке Lazarus не первая и есть четкое понимание данного языка.

Таким образом, домашняя контрольная работа №8 была выполнена мной в полном объеме.