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»**

**«МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ»**

Выполнил: студент учебной группы

ИСПк-202-52-00

Подпорин А.А.

Преподаватель:

Сергеева Е.Г.

Киров

2025

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

**Вариант: 12**

**Задание:**

1. Реализовать приложение "Таймер".
2. Предусмотреть возможность подачи звукового сигнала по истечении заданного промежутка времени.
3. Предусмотреть возможность досрочной остановки таймера.

**Элементы приложения:**

TEdit

Поле для ввода текста пользователем.  
В моем коде: edtHours, edtMinutes, edtSeconds — пользователь вводит время.

TLabel

Отображает текст (не редактируется пользователем).  
В моем коде: Label1, Label2, Label3, lblTime — показывают подписи и оставшееся время.

TButton

Кнопка, при нажатии которой выполняется код.  
В моем коде: btnStart запускает таймер, btnStop — останавливает.

TTimer

Генерирует события через заданные интервалы (в миллисекундах).  
В моем коде: Timer1 обновляет время каждую секунду (каждые 1000 мс).

TForm

Главное окно приложения, которое содержит все остальные элементы (Edit, Button и т.д.).  
В моем коде: Form1 — основное окно таймера.

**Код программы:**

unit Timer;

{$mode objfpc}{$H+}

interface

uses

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

MMSystem; // Для звукового сигнала в Windows

type

{ TForm1 }

TForm1 = class(TForm)

btnStart: TButton;

btnStop: TButton;

edtHours: TEdit;

edtMinutes: TEdit;

edtSeconds: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Time: TLabel;

lblTime: TLabel;

Timer1: TTimer;

procedure btnStartClick(Sender: TObject);

procedure btnStopClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

private

FRemainingTime: Integer;

FIsRunning: Boolean;

procedure UpdateDisplay;

procedure PlaySound;

public

end;

var

Form1: TForm1;

implementation

{$R \*.lfm}

{ TForm1 }

procedure TForm1.FormCreate(Sender: TObject);

begin

FIsRunning := False;

FRemainingTime := 0;

UpdateDisplay;

btnStop.Enabled := False;

end;

procedure TForm1.btnStartClick(Sender: TObject);

var

Hours, Minutes, Seconds: Integer;

begin

if FIsRunning then Exit;

try

if (Trim(edtHours.Text) = '') or (Trim(edtMinutes.Text) = '') or (Trim(edtSeconds.Text) = '') then

raise Exception.Create('Заполните все поля!');

Hours := StrToInt(edtHours.Text);

Minutes := StrToInt(edtMinutes.Text);

Seconds := StrToInt(edtSeconds.Text);

if (Hours < 0) or (Minutes < 0) or (Seconds < 0) then

raise Exception.Create('Время не может быть отрицательным');

if (Minutes >= 60) or (Seconds >= 60) then

raise Exception.Create('Минуты и секунды должны быть меньше 60');

FRemainingTime := Hours \* 3600 + Minutes \* 60 + Seconds;

if FRemainingTime = 0 then

raise Exception.Create('Установите время больше 0');

FIsRunning := True;

btnStart.Enabled := False;

btnStop.Enabled := True;

Timer1.Enabled := True;

except

on E: Exception do

ShowMessage('Ошибка: ' + E.Message);

end;

end;

procedure TForm1.btnStopClick(Sender: TObject);

begin

FIsRunning := False;

Timer1.Enabled := False;

btnStart.Enabled := True;

btnStop.Enabled := False;

end;

procedure TForm1.Timer1Timer(Sender: TObject);

begin

if not FIsRunning then Exit;

if FRemainingTime <= 0 then

begin

Timer1.Enabled := False;

FIsRunning := False;

btnStart.Enabled := True;

btnStop.Enabled := False;

PlaySound;

ShowMessage('Время вышло!');

Exit;

end;

Dec(FRemainingTime);

UpdateDisplay;

end;

procedure TForm1.UpdateDisplay;

var

Hours, Minutes, Seconds: Integer;

begin

Hours := FRemainingTime div 3600;

Minutes := (FRemainingTime mod 3600) div 60;

Seconds := FRemainingTime mod 60;

lblTime.Caption := Format('%.2d:%.2d:%.2d', [Hours, Minutes, Seconds]);

end;

procedure TForm1.PlaySound;

begin

// Для Windows

{$IFDEF WINDOWS}

sndPlaySound('SystemExclamation', SND\_ASYNC);

{$ENDIF}

end;

end.