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

Сосновский Никита Владимирович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

Вариант 19

**Задание:**

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

**Описание алгоритма**

1. Пользователь вводит время в минутах и секундах в поля ввода Edit1 и Edit2.
2. Пользователь нажимает на кнопку "Пуск", что запускает таймер.
3. Таймер начинает отсчет времени каждую секунду.
4. Если время истекло, то воспроизводится звуковой файл и выводится сообщение об окончании времени.
5. Пользователь может остановить таймер, нажав на кнопку "Стоп".
6. Пользователь может снова ввести новое время и запустить таймер снова.

**Результат выполнения программы**

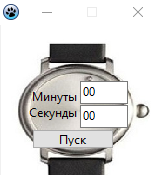


Рисунок 1 – выбор времени

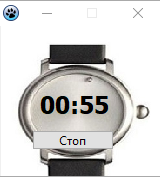


Рисунок 2 – отсчет пошел

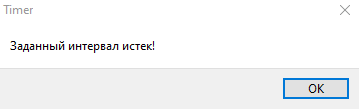


Рисунок 3 – сообщение при окончании времени

**Исходный код**

unit main;

{$mode objfpc}{$H+}

interface

uses

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

type

{ TForm1 }

TForm1 = class(TForm)

Button1: TButton;

Edit1: TEdit;

Edit2: TEdit;

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Timer1: TTimer;

procedure Button1Click(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

private

public

end;

var

Form1: TForm1;

implementation

uses MMSystem;

{$R \*.lfm}

var

min, sec: integer;

procedure ShowTime;

var

buf: String[10];

begin

// минуты и секунды выводим двумя цифрами

if min < 10 then buf := '0' + IntToStr(min) + ':'

else buf := IntToStr(min) + ':';

if sec < 10 then buf := buf + '0' + IntToStr(sec)

else buf := buf + IntToStr(sec);

Form1.Label3.Caption := buf;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

// если таймер работает, надо остановить

if Form1.Timer1.Enabled then begin

Timer1.Enabled := False; // остановить

Button1.Caption := 'Пуск';

// скрыть время, сделать видимыми поля ввода интервала

Label3.Visible:= False;

Label1.Visible:= True; Edit1.Visible:= True;

Label2.Visible:= True; Edit2.Visible:= True;

end

else begin

// таймер стоит, надо запустить

min := StrToInt(Edit1.Text); sec := StrToInt(Edit2.Text);

if (sec = 0) and (min = 0) then begin

ShowMessage('Надо задать интервал!');

exit;

end;

Timer1.Enabled := True; Button1.Caption := 'Стоп';

// скрыть поля ввода интервала

Label1.Visible := False; Edit1.Visible := False;

Label2.Visible := False; Edit2.Visible := False;

Label3.Visible := True;

ShowTime;

end;

end;

procedure TForm1.Timer1Timer(Sender: TObject);

begin

sec := sec - 1;

if sec < 0 then sec := 0;

ShowTime; // показать время

// заданный интервал истек

if (min = 0) and (sec = 0) then begin

Timer1.Enabled := False;

Button1.Caption := 'Пуск';

// Воспроизведение звукового файла

sndPlaySound('chin-chan-chon-chi-chicha-chochi.wav', snd\_Async or snd\_NoDefault);

ShowMessage('Вемя истекло!');

sndPlaySound(nil, 0);

exit;

Label3.Visible := False; // скрыть время

Label1.Visible := True;

Edit1.Visible := True;

Label2.Visible := True;

Edit2.Visible := True;

exit;

end;

if (sec = 0) and (min > 0) then begin

sec := 60;

min := min - 1;

end;

end;

end.

**Вывод по работе**

Таким образом, в данной работе использовался компонент таймер в среде Lazarus. С помощью таймера (Timer) можно запрограммировать выполнение определенного кода через равные интервалы времени. Когда таймер установлен на форме, система периодически генерирует событие OnTimer. Для пользователя таймер невидим.

Данная работа посвящена реализации секундомера — это тип [часов](https://en.wikipedia.org/wiki/Clock), которые запускаются с заданной продолжительности времени и останавливаются по достижении 00:00.

В ходе выполнения работы были встречены проблемы, такие как переключение видов таймера (выбор времени и непосредственно отсчет до 0).

Домашняя контрольная работа №8 была выполнена в полном объёме.