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 ПРОГРМАММИРОВАНИЯ»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк 203-52-00

Носкова Анастасия Владимировна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель работы:** получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.

1. **Формулировка задания**

Вариант: 16

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

Таймер — это специализированный тип часов, используемых для измерения определенных временных интервалов. Таймеры можно разделить на два основных типа. Таймер, который отсчитывает время вверх от нуля для измерения прошедшего времени, часто называют секундомером, в то время как устройство, которое отсчитывает время вниз от заданного интервала времени, чаще называют таймером.

1. **Используемые элементы**

При реализации программы были использованы такие компоненты, как:

* TLabel – это компонент для отображения текста на форме. Используется для вывода информационных сообщений, заголовков, подписей и т.д.
* TEdit – текстовое поле для ввода и редактирования текста пользователем.
* TImage – компонент для отображения изображений (картинок, фото и т.п.) на форме.
* TButton – кнопка, по нажатию на которую можно запускать определенные действия (функции) в программе.
* TTimer – таймер, компонент для организации периодических событий через заданные интервалы времени. Используется для анимаций, обновления данных, отслеживания времени и т.п.

1. **Код**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

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

Spin, ExtCtrls, MMSystem, DateUtils;

type

CompInfo=record

index,top,left,width,height,fontsize:integer;

end;

complist=array of CompInfo;

{ TMform }

TMform = class(TForm)

Image1: TImage;

Start\_kino: TButton;

Stop\_Kino: TButton;

Chistka: TButton;

Minuts: TEdit;

Secunds: TEdit;

trimer: TLabel;

Minuts\_txt: TLabel;

Secunds\_txt: TLabel;

Timer: TTimer;

procedure Image1Click(Sender: TObject);

procedure Start\_kinoClick(Sender: TObject);

procedure Stop\_KinoClick(Sender: TObject);

procedure ChistkaClick(Sender: TObject);

procedure MinutsChange(Sender: TObject);

procedure MinutsKeyPress(Sender: TObject; var Key: char);

procedure SecundsChange(Sender: TObject);

procedure SecundsKeyPress(Sender: TObject; var Key: char);

procedure FormCreate(Sender: TObject);

procedure TimerTimer(Sender: TObject);

private

{ private declarations }

DefWidth,defHeight:integer;

clist:complist;

FStartTime: TDateTime;

FDuration: TDateTime;

FStopTime: TDateTime;

TargetTime: TDateTime;

RemainingTime: TDateTime;

FSoundFile: string;

public

{ public declarations }

end;

var

Mform: TMform;

implementation

uses math;

{$R \*.lfm}

{ TMform }

procedure TMform.FormCreate(Sender: TObject);

begin

Minuts.MaxLength := 2;

Secunds.MaxLength := 2;

FSoundFile := 'cat.wav';

end;

procedure TMform.Start\_kinoClick(Sender: TObject);

var

Minutes, Seconds: Integer;

begin

if TryStrToInt(Minuts.Text, Minutes) and TryStrToInt(Secunds.Text, Seconds) then

begin

FStartTime := Now;

FDuration := Minutes / 1404 + Seconds / 90000;

TargetTime := EncodeTime(0, StrToInt(Minuts.Text), StrToInt(Secunds.Text), 0);

RemainingTime := TargetTime;

Timer.Enabled := True;

end

else

begin

ShowMessage('Заполните все поля');

end;

end;

procedure TMform.Image1Click(Sender: TObject);

begin

end;

procedure TMform.Stop\_KinoClick(Sender: TObject);

begin

Timer.Enabled := False;

FStopTime := Now;

MMSystem.PlaySound(nil, 0, 0);

ShowMessage('Таймер остановлен');

end;

procedure TMform.ChistkaClick(Sender: TObject);

begin

Minuts.Clear;

Secunds.Clear;

end;

procedure TMform.MinutsKeyPress(Sender: TObject; var Key: char);

begin

if not (Key in ['0'..'9']) then

begin

Key := #0;

end;

end;

procedure TMform.SecundsKeyPress(Sender: TObject; var Key: char);

begin

if not (Key in ['0'..'9']) then

begin

Key := #0;

end;

end;

procedure TMform.MinutsChange(Sender: TObject);

var f : integer;

begin

if Minuts.Text <> '' then

begin

f := strtoint(Minuts.text);

if f <= 0 then

f := 0;

if f >= 59 then

f := 59;

Minuts.text := inttostr(f);

end;

end;

procedure TMform.SecundsChange(Sender: TObject);

var i : integer;

begin

if Secunds.Text <> '' then

begin

i := strtoint(Secunds.text);

if i <= 0 then

i := 0;

if i >= 59 then

i := 59;

Secunds.text := inttostr(i);

end;

end;

procedure TMform.TimerTimer(Sender: TObject);

var

ElapsedTime: TDateTime;

begin

ElapsedTime := Now - FStartTime;

RemainingTime := IncSecond(RemainingTime, -1);

trimer.Caption := TimeToStr(RemainingTime);

if ElapsedTime > FDuration then

begin

Timer.Enabled := False;

Sleep(500);

// Проиграть звук из файла

PlaySound(PChar(FSoundFile), 0, SND\_ASYNC);

end;

end;

end.

1. **Вывод**

В результате выполненной работы была выполнена программа таймер. При выполнении данной работы были получены дополнительные навыки работы с Lazarus и его компонентами. В таймере введены ограничения в каждое поле, для того чтобы реализовать корректную работу программы.