Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №9**

**«Работа с типизированными файлами»**

**МДК 05.02 «Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-202-52-00

Гончаров Мирослав Даниилович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Цель работы**

Получить представление об организации хранения и использования информации посредством типизированных файлов, закрепить навыки создания пользовательского интерфейса.

**Задание для варианта 4**

1. Сформировать структуру записи, с которой будет осуществляться работа, и согласовать её с преподавателем.
2. Разработать приложение, позволяющее извлекать набор записей из типизированного файла, визуализировать данный набор, а также позволяющее изменять данные и выполнять сохранение проделанных изменений.
3. В записи должно быть не менее 5 элементов, среди которых должно быть минимум 3 различных типа данных.

# **Описание алгоритма**

На главной форме расположены элементы управления для работы с записями философов:

Элементы интерфейса:

* **TBitBtn** - кнопки для добавления (**Add\_Note**), редактирования (**Edit\_Note**) и удаления (**Del\_Note**) записей.
* **TPanel** - панель для размещения кнопок управления.
* **TStringGrid** - таблица для отображения записей философов (**mRec**).

Основной алгоритм работы:

1. **Инициализация формы (FormCreate)**

При создании формы инициализируются элементы управления и заполняется заголовок таблицы.

Если файл данных существует, считываются данные из файла и заполняется таблица.

1. **Добавление записи (Add\_NoteClick)**

Очищаются поля ввода формы **createRecord**.

Отображается форма для ввода новой записи.

Если введены все необходимые данные и нажата кнопка "Сохранить", новая запись добавляется в таблицу.

1. **Удаление записи (Del\_NoteClick)**

Если в таблице есть данные, отображается диалог подтверждения удаления.

При подтверждении выбранная запись удаляется из таблицы.

1. **Редактирование записи (Edit\_NoteClick)**

Если в таблице есть данные, выбранная запись копируется в поля формы **createRecord**.

Отображается форма для редактирования записи.

При сохранении изменений запись обновляется в таблице.

1. **Закрытие формы (FormClose)**

Если в таблице есть данные, они записываются в файл при закрытии формы.

# **Код программы**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

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

type

{ TFind\_Figure\_Square }

TFind\_Figure\_Square = class(TForm)

Select\_Figure: TRadioGroup;

Formula\_Picture: TImage;

First\_val\_for\_calcs: TEdit;

Second\_val\_for\_calcs: TEdit;

Third\_val\_for\_calcs: TEdit;

Calculate\_Square: TButton;

Answer\_Field: TMemo;

Label\_for\_first\_val: TLabel;

Label\_for\_second\_val: TLabel;

Label\_for\_third\_val: TLabel;

procedure FormCreate(Sender: TObject);

procedure Answer\_FieldClick(Sender: TObject);

procedure Select\_FigureClick(Sender: TObject);

procedure Calculate\_SquareClick(Sender: TObject);

private

procedure UpdateUIForSelectedFigure;

function IsNumber(const S: string): Boolean;

public

end;

var

Find\_Figure\_Square: TFind\_Figure\_Square;

implementation

{$R \*.lfm}

{ TFind\_Figure\_Square }

procedure TFind\_Figure\_Square.Answer\_FieldClick(Sender: TObject);

begin

end;

procedure TFind\_Figure\_Square.FormCreate(Sender: TObject);

begin

end;

procedure TFind\_Figure\_Square.Select\_FigureClick(Sender: TObject);

begin

First\_val\_for\_calcs.Text := '';

Second\_val\_for\_calcs.Text := '';

Third\_val\_for\_calcs.Text := '';

Answer\_Field.Text := '';

Find\_Figure\_Square.Height := 550;

UpdateUIForSelectedFigure;

end;

procedure TFind\_Figure\_Square.UpdateUIForSelectedFigure;

begin

// Скрываем все компоненты

Formula\_Picture.Visible := False;

First\_val\_for\_calcs.Visible := False;

Second\_val\_for\_calcs.Visible := False;

Third\_val\_for\_calcs.Visible := False;

Calculate\_Square.Visible := False;

Answer\_Field.Visible := False;

Label\_for\_first\_val.Visible := False;

Label\_for\_second\_val.Visible := False;

Label\_for\_third\_val.Visible := False;

// Обновляем UI в зависимости от выбранной фигуры

case Select\_Figure.ItemIndex of

0: // Круг

begin

Formula\_Picture.Picture.LoadFromFile('square\_circles.png');

Formula\_Picture.Visible := True;

First\_val\_for\_calcs.Hint := 'Радиус';

First\_val\_for\_calcs.Visible := True;

Calculate\_Square.Visible := True;

Label\_for\_first\_val.Caption := 'R =';

Label\_for\_first\_val.Visible := True;

end;

1: // Прямоугольник

begin

Formula\_Picture.Picture.LoadFromFile('square\_rectangles.png');

Formula\_Picture.Visible := True;

First\_val\_for\_calcs.Hint := 'Длина';

Second\_val\_for\_calcs.Hint := 'Ширина';

First\_val\_for\_calcs.Visible := True;

Second\_val\_for\_calcs.Visible := True;

Calculate\_Square.Visible := True;

Label\_for\_first\_val.Caption := 'a =';

Label\_for\_second\_val.Caption := 'b =';

Label\_for\_first\_val.Visible := True;

Label\_for\_second\_val.Visible := True;

end;

2: // Треугольник

begin

Formula\_Picture.Picture.LoadFromFile('square\_triangle.png');

Formula\_Picture.Visible := True;

First\_val\_for\_calcs.Hint := 'Основание';

Second\_val\_for\_calcs.Hint := 'Высота';

First\_val\_for\_calcs.Visible := True;

Second\_val\_for\_calcs.Visible := True;

Calculate\_Square.Visible := True;

Label\_for\_first\_val.Caption := 'a =';

Label\_for\_second\_val.Caption := 'h =';

Label\_for\_first\_val.Visible := True;

Label\_for\_second\_val.Visible := True;

end;

3: // Трапеция

begin

Formula\_Picture.Picture.LoadFromFile('square\_trapezoid.png');

Formula\_Picture.Visible := True;

First\_val\_for\_calcs.Hint := 'Основание 1';

Second\_val\_for\_calcs.Hint := 'Основание 2';

Third\_val\_for\_calcs.Hint := 'Высота';

First\_val\_for\_calcs.Visible := True;

Second\_val\_for\_calcs.Visible := True;

Third\_val\_for\_calcs.Visible := True;

Calculate\_Square.Visible := True;

Label\_for\_first\_val.Caption := 'a =';

Label\_for\_second\_val.Caption := 'b =';

Label\_for\_third\_val.Caption := 'h =';

Label\_for\_first\_val.Visible := True;

Label\_for\_second\_val.Visible := True;

Label\_for\_third\_val.Visible := True;

end;

end;

end;

function TFind\_Figure\_Square.IsNumber(const S: string): Boolean;

var

TempStr: string;

Regex: TRegExpr;

begin

TempStr := StringReplace(S, ',', '.', [rfReplaceAll]); // Заменяем запятые на точки для проверки

Regex := TRegExpr.Create;

try

Regex.Expression := '^[0-9]+([.,][0-9]+)?$'; // Регулярное выражение для проверки числового значения с точкой или запятой

Result := Regex.Exec(TempStr);

finally

Regex.Free;

end;

end;

procedure TFind\_Figure\_Square.Calculate\_SquareClick(Sender: TObject);

var

Area, Value1, Value2, Value3: Double;

begin

if First\_val\_for\_calcs.Visible then

begin

if (First\_val\_for\_calcs.Text = '') or not IsNumber(First\_val\_for\_calcs.Text) or (StrToFloat(StringReplace(First\_val\_for\_calcs.Text, '.', ',', [rfReplaceAll])) < 0) then

begin

ShowMessage('Пожалуйста, введите корректное значение для ' + First\_val\_for\_calcs.Hint);

Exit;

end;

end;

if Second\_val\_for\_calcs.Visible then

begin

if (Second\_val\_for\_calcs.Text = '') or (not IsNumber(Second\_val\_for\_calcs.Text)) or (StrToFloat(StringReplace(Second\_val\_for\_calcs.Text, '.', ',', [rfReplaceAll])) < 0) then

begin

ShowMessage('Пожалуйста, введите корректное значение для ' + Second\_val\_for\_calcs.Hint);

Exit;

end;

end;

if Third\_val\_for\_calcs.Visible then

begin

if (Third\_val\_for\_calcs.Text = '') or not IsNumber(Third\_val\_for\_calcs.Text) or (StrToFloat(StringReplace(Third\_val\_for\_calcs.Text, '.', ',', [rfReplaceAll])) < 0) then

begin

ShowMessage('Пожалуйста, введите корректное значение для ' + Third\_val\_for\_calcs.Hint);

Exit;

end;

end;

Value1 := StrToFloatDef(StringReplace(First\_val\_for\_calcs.Text, '.', ',', [rfReplaceAll]),0);

Value2 := StrToFloatDef(StringReplace(Second\_val\_for\_calcs.Text, '.', ',', [rfReplaceAll]),0);

Value3 := StrToFloatDef(StringReplace(Third\_val\_for\_calcs.Text, '.', ',', [rfReplaceAll]),0);

case Select\_Figure.ItemIndex of

0: // Круг

Area := Pi \* Sqr(Value1);

1: // Прямоугольник

Area := Value1 \* Value2;

2: // Треугольник

Area := 0.5 \* Value1 \* Value2;

3: // Трапеция

Area := 0.5 \* (Value1 + Value2) \* Value3;

end;

Find\_Figure\_Square.Height := 700;

Answer\_Field.Visible := True;

Answer\_Field.Lines.Add('Рассчитанная площадь: ' + FloatToStr(Area));

end;

end.

# **Результат выполнения программы**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рис. 1 – Результат выполнения программы

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

Автоматически созданное описание

Рис. 2 – Результат выполнения программы

Изображение выглядит как текст, снимок экрана, дисплей, Шрифт

Автоматически созданное описание

Рис. 3 – Результат выполнения программы

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

Автоматически созданное описание

Рис. 4 – Результат выполнения программы

# **Вывод**

В процессе выполнения данной лабораторной работы была достигнута основная цель: получение навыков организации хранения и использования информации посредством типизированных файлов, а также закрепление навыков создания пользовательского интерфейса. Работа включала в себя разработку программы на языке Delphi в среде Lazarus, которая позволяла извлекать набор записей из типизированного файла, визуализировать данный набор, изменять данные и сохранять произведённые изменения. В записи содержалось не менее пяти элементов, среди которых было минимум три различных типа данных. Было получено представление об организации хранения и использования информации посредством типизированных файлов, закреплены навыки создания пользовательского интерфейса.