УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №4.1

по предмету «Основы алгоритмизации и программирования»

Вариант 4

Выполнил:

Бражалович А. И.

Гр. 351004

Проверил:

Данилова Г. В.

Минск 2024

**Задание:**

Сведения о расписании занятий:

* день недели;
* номер пары;
* название предмета;
* фамилия преподавателя.

Вывести: сведения о расписании занятий каждого преподавателя по дням недели.

В программе реализованы следующие функции:

* ввод записей;
* корректировка записей;
* удаление записей;
* просмотр записей на экране;
* отбор информации по заданному критерию.

Требуемая функция выбирается с помощью меню.

**Код программы Delphi:**

Unit Unit41;

Interface

Uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Grids, Vcl.StdCtrls, Unit14,

Unit155,

Vcl.Buttons, Vcl.Menus, Vcl.ExtDlgs;

Type

TScheduleInfo = Record

ElemSubject: String;

ElemTeacher: String;

ElemDayOfWeek: String;

ElemNumOfLesson: Byte;

End;

TScheduleInfoArr = Array Of TScheduleInfo;

StringGridCracker = Class(TStringGrid);

TForm13 = Class(TForm)

Label1: TLabel;

Button1: TButton;

TeacherSch: TButton;

Records: TStringGrid;

SpeedButton1: TSpeedButton;

MainMenu1: TMainMenu;

N1: TMenuItem;

BOpen: TMenuItem;

BSave: TMenuItem;

BClose: TMenuItem;

Instructions: TMenuItem;

BDev: TMenuItem;

DeleteRec: TButton;

OpenTextFileDialog1: TOpenTextFileDialog;

ShiftBlocker: TMenuItem;

Procedure FormCreate(Sender: TObject);

Procedure Button1Click(Sender: TObject);

Procedure TeacherSchClick(Sender: TObject);

Procedure MyCellClick(Sender: TObject);

Procedure RecordsSelectCell(Sender: TObject; ACol, ARow: Integer;

Var CanSelect: Boolean);

Procedure DeleteRecClick(Sender: TObject);

Procedure BDevClick(Sender: TObject);

Procedure FormCloseQuery(Sender: TObject; Var CanClose: Boolean);

Procedure BSaveClick(Sender: TObject);

Procedure BOpenClick(Sender: TObject);

procedure InstructionsClick(Sender: TObject);

function FormHelp(Command: Word; Data: NativeInt;

var CallHelp: Boolean): Boolean;

procedure BCloseClick(Sender: TObject);

Private

Public

End;

Var

Form13: TForm13;

Implementation

{$R \*.dfm}

procedure TForm13.BCloseClick(Sender: TObject);

begin

Close

end;

Procedure TForm13.BDevClick(Sender: TObject);

Begin

Application.MessageBox('Бражалович Александр Иванович,' + #13 + 'группа 351004',

'О разработчике')

End;

Function IsFileReadOnly(Const FileName: String): Boolean;

Begin

Result := False;

If FileExists(FileName) Then

Begin

Result := (FileGetAttr(FileName) And FaReadOnly) = FaReadOnly;

End;

End;

Procedure TForm13.BOpenClick(Sender: TObject);

Var

TxtFile: TextFile;

CanReadFile: Boolean;

FileName: String;

CurNum, NumOfIter, I, J, K: Integer;

RecordsFromFile, MyRecords: TScheduleInfoArr;

MyRecordFromFile, CurRecord: TScheduleInfo;

Begin

CanReadFile := True;

OpenTextFileDialog1.Filter := 'Текстовый файл (\*.txt)|\*.txt';

If OpenTextFileDialog1.Execute Then

Begin

FileName := OpenTextFileDialog1.FileName;

AssignFile(TxtFile, FileName);

Try

Reset(TxtFile);

Except

Application.MessageBox('Не удалось открыть файл', 'Error');

CanReadFile := False;

End;

If Not SameText(ExtractFileExt(FileName), '.txt') And CanReadFile Then

Begin

CanReadFile := False;

Application.MessageBox('Неверное расширение файла', 'Error');

End;

If CanReadFile Then

Begin

SetLength(MyRecords, Records.RowCount - 1);

If Records.RowCount > 1 Then

For I := 0 To Records.RowCount - 2 Do

Begin

MyRecords[I].ElemSubject := Records.Cells[2, I + 1];

MyRecords[I].ElemTeacher := Records.Cells[3, I + 1];

MyRecords[I].ElemDayOfWeek := Records.Cells[0, I + 1];

MyRecords[I].ElemNumOfLesson :=

StrToInt(Records.Cells[1, I + 1]);

End;

Try

If EOF(TxtFile) Then

Begin

Application.MessageBox('Число записей не задано', 'Error');

CanReadFile := False;

End

Else If CanReadFile Then

Begin

Readln(TxtFile, NumOfIter);

If (NumOfIter > 512) And (NumOfIter < 0) Then

Begin

Button1.Enabled := False;

Application.MessageBox

('Число вне диапазона, F1 для помощи', 'Error');

CanReadFile := False

End

Else

SetLength(RecordsFromFile, NumOfIter)

End;

For I := 0 To NumOfIter - 1 Do

Begin

For J := 1 To 4 Do

Begin

If EOF(TxtFile) And CanReadFile Then

Begin

Application.MessageBox

('Не все элементы введены', 'Error');

CanReadFile := False

End

Else If CanReadFile Then

Begin

Case J Mod 4 Of

0:

Begin

Readln(TxtFile,

MyRecordFromFile.ElemTeacher);

If Length(MyRecordFromFile.

ElemTeacher) > 14 Then

Begin

Application.MessageBox

('Не корректное заполнение файла, F1 для помощи', 'Error');

CanReadFile := False;

End;

End;

1:

Begin

Readln(TxtFile,

MyRecordFromFile.ElemDayOfWeek);

If Not(MyRecordFromFile.ElemDayOfWeek =

'Понедельник') And

Not(MyRecordFromFile.ElemDayOfWeek =

'Вторник') And

Not(MyRecordFromFile.ElemDayOfWeek =

'Среда') And

Not(MyRecordFromFile.ElemDayOfWeek =

'Четверг') And

Not(MyRecordFromFile.ElemDayOfWeek =

'Пятница') And

Not(MyRecordFromFile.ElemDayOfWeek =

'Суббота') Then

Begin

Application.MessageBox

('Не корректное заполнение файла, F1 для помощи', 'Error');

CanReadFile := False;

End;

End;

2:

Begin

Readln(TxtFile,

MyRecordFromFile.ElemNumOfLesson);

If Not(MyRecordFromFile.ElemNumOfLesson In [1..6]) Then

Begin

Application.MessageBox

('Не корректное заполнение файла, F1 для помощи', 'Error');

CanReadFile := False;

End;

End;

3:

Begin

Readln(TxtFile,

MyRecordFromFile.ElemSubject);

If Length(MyRecordFromFile.

ElemTeacher) > 14 Then

Begin

Application.MessageBox

('Не корректное заполнение файла, F1 для помощи', 'Error');

CanReadFile := False;

End;

End;

End;

End;

End;

If CanReadFile Then

Begin

For K := 0 To High(MyRecords) Do

Begin

CurRecord := MyRecords[K];

If (CurRecord.ElemTeacher = MyRecordFromFile.

ElemTeacher) And

(CurRecord.ElemDayOfWeek = MyRecordFromFile.

ElemDayOfWeek) And

(CurRecord.ElemNumOfLesson = MyRecordFromFile.

ElemNumOfLesson) Then

Begin

Application.MessageBox

('Один из элементов уже присутствует в записях',

'Error');

CanReadFile := False

End

End;

For K := 0 To I - 1 Do

Begin

CurRecord := RecordsFromFile[K];

If (CurRecord.ElemTeacher = MyRecordFromFile.

ElemTeacher) And

(CurRecord.ElemDayOfWeek = MyRecordFromFile.

ElemDayOfWeek) And

(CurRecord.ElemNumOfLesson = MyRecordFromFile.

ElemNumOfLesson) Then

Begin

Application.MessageBox

('Запись в файле повторяется', 'Error');

CanReadFile := False

End

End;

End;

If CanReadFile Then

RecordsFromFile[I] := MyRecordFromFile;

End;

If CanReadFile Then

Begin

For I := 0 To NumOfIter - 1 Do

Begin

Records.RowCount := Records.RowCount + 1;

Records.Cells[0, Records.RowCount - 1] :=

RecordsFromFile[I].ElemDayOfWeek;

Records.Cells[1, Records.RowCount - 1] :=

IntToStr(RecordsFromFile[I].ElemNumOfLesson);

Records.Cells[2, Records.RowCount - 1] :=

RecordsFromFile[I].ElemSubject;

Records.Cells[3, Records.RowCount - 1] :=

RecordsFromFile[I].ElemTeacher;

End;

End;

Except

Application.MessageBox

('Не корректное заполнение файла, F1 для помощи', 'Error');

CanReadFile := False;

End;

End;

CloseFile(TxtFile);

If Records.RowCount > 1 Then

BSave.Enabled := True

Else

BSave.Enabled := False

End;

End;

Procedure TForm13.BSaveClick(Sender: TObject);

Var

I, J: Integer;

TxtFile: TextFile;

CanReadFile, WantSave: Boolean;

FileName: String;

CurNum: Double;

Begin

CanReadFile := True;

OpenTextFileDialog1.Filter := 'Текстовый файл (\*.txt)|\*.txt';

If OpenTextFileDialog1.Execute Then

Begin

FileName := OpenTextFileDialog1.FileName;

If IsFileReadOnly(FileName) Then

Begin

Application.MessageBox('Файл не открыт для чтения', 'Error');

CanReadFile := False;

End;

If Not SameText(ExtractFileExt(FileName), '.txt') And CanReadFile Then

Begin

CanReadFile := True;

AssignFile(TxtFile, FileName + '.txt');

Try

Rewrite(TxtFile);

Except

Application.MessageBox('Не удалось открыть файл', 'Error');

CanReadFile := False;

End;

End

Else If CanReadFile Then

Begin

AssignFile(TxtFile, FileName);

Try

Rewrite(TxtFile);

Except

Application.MessageBox('Не удалось открыть файл', 'Error');

CanReadFile := False;

End;

End;

If CanReadFile Then

Begin

Writeln(TxtFile, IntToStr(Records.RowCount - 1));

For J := 1 To Records.RowCount - 1 Do

Begin

Writeln(TxtFile, Records.Cells[0, J]);

Writeln(TxtFile, Records.Cells[1, J]);

Writeln(TxtFile, Records.Cells[2, J]);

Writeln(TxtFile, Records.Cells[3, J]);

End;

End;

CloseFile(TxtFile);

End;

End;

Procedure TForm13.Button1Click(Sender: TObject);

Var

NewForm: TForm14;

I: Integer;

Begin

DeleteRec.Enabled := False;

I := 1;

NewForm := TForm14.Create(Application);

Try

NewForm.WantEdit := False;

NewForm.ShowModal;

While (I < Records.RowCount) And

(NewForm.CurRecord.ElemSubject <> '') Do

Begin

If (Records.Cells[0, I] = NewForm.CurRecord.ElemDayOfWeek) And

(Records.Cells[1, I] = IntToStr(NewForm.CurRecord.ElemNumOfLesson)

) And (Records.Cells[3, I] = NewForm.CurRecord.ElemTeacher) Then

Begin

NewForm.CurRecord.ElemSubject := '';

Application.MessageBox

('Данная дата и занятие у выбранного пеподователя уже заняты',

'Предупреждение');

End;

Inc(I);

End;

If NewForm.CurRecord.ElemSubject <> '' Then

Begin

Records.RowCount := Records.RowCount + 1;

Records.Cells[0, Records.RowCount - 1] :=

NewForm.CurRecord.ElemDayOfWeek;

Records.Cells[1, Records.RowCount - 1] :=

IntToStr(NewForm.CurRecord.ElemNumOfLesson);

Records.Cells[2, Records.RowCount - 1] :=

NewForm.CurRecord.ElemSubject;

Records.Cells[3, Records.RowCount - 1] :=

NewForm.CurRecord.ElemTeacher;

End;

Finally

NewForm.Free;

End;

If Records.RowCount = 1 Then

Begin

TeacherSch.Enabled := False;

BSave.Enabled := False;

End

Else

Begin

TeacherSch.Enabled := True;

BSave.Enabled := True;

End;

End;

Procedure TForm13.TeacherSchClick(Sender: TObject);

Var

NewForm: TForm15;

I: Integer;

Begin

DeleteRec.Enabled := False;

NewForm := TForm15.Create(Application);

Try

SetLength(NewForm.AllRecords, Records.RowCount - 1);

For I := 1 To Records.RowCount - 1 Do

Begin

NewForm.AllRecords[I - 1].ElemDayOfWeek := Records.Cells[0, I];

NewForm.AllRecords[I - 1].ElemNumOfLesson :=

StrToInt(Records.Cells[1, I]);

NewForm.AllRecords[I - 1].ElemSubject := Records.Cells[2, I];

NewForm.AllRecords[I - 1].ElemTeacher := Records.Cells[3, I];

End;

NewForm.ShowModal;

Finally

NewForm.Free;

End;

End;

Procedure TForm13.DeleteRecClick(Sender: TObject);

Var

I: Integer;

Begin

With CreateMessageDialog('Вы действительно хотите удалить запись?',

MtConfirmation, MbYesNo) Do

Try

Caption := 'Выход';

TButton(FindComponent('Yes')).Caption := 'Да';

TButton(FindComponent('No')).Caption := 'Нет';

Case ShowModal Of

MrYes:

Begin

For I := Records.Row To Records.RowCount - 2 Do

Begin

Records.Cells[0, I] := Records.Cells[0, I + 1];

Records.Cells[1, I] := Records.Cells[1, I + 1];

Records.Cells[2, I] := Records.Cells[2, I + 1];

Records.Cells[3, I] := Records.Cells[3, I + 1];

End;

Records.RowCount := Records.RowCount - 1;

If Records.RowCount = 1 Then

Begin

TeacherSch.Enabled := False;

BSave.Enabled := False;

End

Else

Begin

TeacherSch.Enabled := True;

BSave.Enabled := True;

End;

End;

End;

Finally

End;

End;

Procedure TForm13.FormCloseQuery(Sender: TObject; Var CanClose: Boolean);

Var

CustomButtons: TMsgDlgButtons;

Free: Boolean;

Begin

CanClose := False;

If Records.RowCount < 2 Then

Begin

With CreateMessageDialog('Вы действительно хотите закрыть программу?',

MtConfirmation, MbYesNo) Do

Try

Caption := 'Выход';

TButton(FindComponent('Yes')).Caption := 'Да';

TButton(FindComponent('No')).Caption := 'Нет';

Case ShowModal Of

MrYes:

CanClose := True;

MrNo:

CanClose := False;

End;

Finally

End;

End

Else

Begin

With CreateMessageDialog('Cохранить перед выходом?', MtConfirmation,

MbYesNoCancel) Do

Try

Caption := 'Выход';

TButton(FindComponent('Yes')).Caption := 'Выход';

TButton(FindComponent('No')).Caption := 'Сохранить';

TButton(FindComponent('Cancel')).Caption := 'Отмена';

Case ShowModal Of

MrYes:

CanClose := True;

MrCancel:

CanClose := False;

MrNo:

Begin

BSaveClick(BSave);

CanClose := False;

End;

End;

Finally

End;

End;

End;

Procedure TForm13.FormCreate(Sender: TObject);

Begin

Records.Cells[0, 0] := 'День недели';

Records.Cells[1, 0] := '№ занятия';

Records.Cells[2, 0] := 'Предмет';

Records.Cells[3, 0] := 'Преподователь';

End;

function TForm13.FormHelp(Command: Word; Data: NativeInt;

var CallHelp: Boolean): Boolean;

begin

CallHelp := False;

end;

procedure TForm13.InstructionsClick(Sender: TObject);

begin

Application.MessageBox('Для добавления записей в таблицу вы можете воспользоваться кнопкой добавить или' + ' кнопкой файл->открыть. Для дополнения таблицы из файла' + ' необходимо в файле в первой'+' строке ввести количество вводимых записей,'+' затем в каждой новой строке нужно вводить новый элемент добавляемой записи'+'(День недели(c большой буквы) -> Номер ' + 'пары(1..6) -> Название предмета(< 15 символов) -> ' + 'Имя преподавателя(< 15 символов).', 'Помощь');

Application.MessageBox('Для изменении'+' записи нажмите на нее в таблице.'+' Для получения расписания преподователя нажмите на кнопку Занятость. Для удаления записи ' + ' нажмите на нее в таблице и нажмите удалить. Для получения помощи нажмите F1', 'Помощь');

end;

Procedure TForm13.MyCellClick(Sender: TObject);

Var

I: Integer;

NewForm: TForm14;

Begin

If Records.Row > 0 Then

Begin

NewForm := TForm14.Create(Application);

Try

NewForm.AddRec.Visible := False;

NewForm.ChangeRecord.Visible := True;

Case Records.Cells[0, Records.Row][1] Of

'П':

Begin

If Records.Cells[0, Records.Row][2] = 'о' Then

NewForm.DayOfWeek.ItemIndex := 0

Else

NewForm.DayOfWeek.ItemIndex := 4

End;

'В':

NewForm.DayOfWeek.ItemIndex := 1;

'С':

Begin

If Records.Cells[0, Records.Row][2] = 'р' Then

NewForm.DayOfWeek.ItemIndex := 2

Else

NewForm.DayOfWeek.ItemIndex := 5

End;

'Ч':

NewForm.DayOfWeek.ItemIndex := 3

End;

NewForm.LessonNum.ItemIndex :=

StrToInt(Records.Cells[1, Records.Row]) - 1;

NewForm.Subject.Text := Records.Cells[2, Records.Row];

NewForm.Teacher.Text := Records.Cells[3, Records.Row];

NewForm.ShowModal;

If NewForm.CurRecord.ElemSubject <> '' Then

Begin

Records.Cells[0, Records.Row] :=

NewForm.CurRecord.ElemDayOfWeek;

Records.Cells[1, Records.Row] :=

IntToStr(NewForm.CurRecord.ElemNumOfLesson);

Records.Cells[2, Records.Row] := NewForm.CurRecord.ElemSubject;

Records.Cells[3, Records.Row] := NewForm.CurRecord.ElemTeacher;

End;

Finally

NewForm.Free;

End;

End;

End;

Procedure TForm13.RecordsSelectCell(Sender: TObject; ACol, ARow: Integer;

Var CanSelect: Boolean);

Var

MyCell: StringGridCracker;

Begin

MyCell := StringGridCracker(Records);

MyCell.OnDblClick := MyCellClick;

If ARow > 0 Then

DeleteRec.Enabled := True

Else

DeleteRec.Enabled := False;

End;

End.

Unit Unit14;

Interface

Uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Menus;

Type

ScheduleInfo = Record

ElemSubject: String;

ElemTeacher: String;

ElemDayOfWeek: String;

ElemNumOfLesson: Byte;

End;

ScheduleInfoArr = Array Of ScheduleInfo;

TForm14 = Class(TForm)

Label1: TLabel;

DayOfWeek: TComboBox;

LessonNum: TComboBox;

Subject: TEdit;

Teacher: TEdit;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

CancelRec: TButton;

AddRec: TButton;

ChangeRecord: TButton;

MainMenu1: TMainMenu;

Instructions: TMenuItem;

PopupMenu1: TPopupMenu;

gg: TMenuItem;

ShiftBlocker: TMenuItem;

Procedure CancelRecClick(Sender: TObject);

Procedure SubjectKeyPress(Sender: TObject; Var Key: Char);

Procedure TeacherKeyPress(Sender: TObject; Var Key: Char);

Procedure SubjectChange(Sender: TObject);

Procedure AddRecClick(Sender: TObject);

procedure InstructionsClick(Sender: TObject);

Private

{ Private declarations }

Public

RecordInfo: ScheduleInfo;

CurRecord: ScheduleInfo;

WantEdit: Boolean;

End;

Var

Form14: TForm14;

Implementation

{$R \*.dfm}

Procedure TForm14.AddRecClick(Sender: TObject);

Begin

CurRecord.ElemSubject := Subject.Text;

CurRecord.ElemTeacher := Teacher.Text;

CurRecord.ElemDayOfWeek := DayOfWeek.Text;

CurRecord.ElemNumOfLesson := StrToInt(LessonNum.Text);

Close

End;

Procedure TForm14.CancelRecClick(Sender: TObject);

Begin

CurRecord.ElemSubject := '';

Close

End;

procedure TForm14.InstructionsClick(Sender: TObject);

begin

Application.MessageBox('В верхние ячейки введите соответствующие строки длинной не больше 15, затем в нижних меню выберите соответствующие значения. Чтобы отменить действие нажмите на кнопку отменить ', 'Помощь');

end;

Procedure TForm14.SubjectChange(Sender: TObject);

Begin

If (Teacher.Text <> '') And (Subject.Text <> '') And (DayOfWeek.Text <> '')

And (LessonNum.Text <> '') Then

AddRec.Enabled := True

Else

AddRec.Enabled := False

End;

Procedure TForm14.SubjectKeyPress(Sender: TObject; Var Key: Char);

Begin

If ((Length(Subject.Text) > 14) And (Key <> #8)) Or

(Key In ['0' .. '9']) Then

Key := #0;

End;

Procedure TForm14.TeacherKeyPress(Sender: TObject; Var Key: Char);

Begin

If ((Length(Teacher.Text) > 14) And (Key <> #8)) Or

(Key In ['0' .. '9']) Then

Key := #0;

End;

End.

Unit Unit155;

Interface

Uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Grids, Vcl.StdCtrls, Vcl.Menus;

Type

ScheduleInfo = Record

ElemSubject: String;

ElemTeacher: String;

ElemDayOfWeek: String;

ElemNumOfLesson: Byte;

End;

ScheduleInfoArr = Array Of ScheduleInfo;

TForm15 = Class(TForm)

Label1: TLabel;

Schedule: TStringGrid;

TeachName: TEdit;

Label2: TLabel;

Fill: TButton;

MainMenu1: TMainMenu;

gg: TMenuItem;

N1: TMenuItem;

PopupMenu1: TPopupMenu;

ShiftBlocker: TMenuItem;

Procedure FormCreate(Sender: TObject);

Procedure TeachNameKeyPress(Sender: TObject; Var Key: Char);

Procedure TeachNameChange(Sender: TObject);

Procedure FillClick(Sender: TObject);

procedure N1Click(Sender: TObject);

Private

Public

AllRecords: ScheduleInfoArr;

End;

Var

Form15: TForm15;

Implementation

{$R \*.dfm}

Procedure TForm15.TeachNameChange(Sender: TObject);

Begin

If (TeachName.Text <> '') Then

Fill.Enabled := True

Else

Fill.Enabled := False

End;

Procedure TForm15.TeachNameKeyPress(Sender: TObject; Var Key: Char);

Begin

If ((Length(TeachName.Text) > 19) And (Key <> #8)) Or

(Key In ['0' .. '9']) Then

Key := #0;

End;

Procedure TForm15.FillClick(Sender: TObject);

Var

I: Integer;

HaveTeacher: Boolean;

J: Integer;

Begin

HaveTeacher := False;

For I := 1 To 6 Do

For J := 1 To 6 Do

Schedule.Cells[I, J] := '';

For I := Low(AllRecords) To High(AllRecords) Do

Begin

Case AllRecords[I].ElemDayOfWeek[1] Of

'П':

Begin

If AllRecords[I].ElemDayOfWeek[2] = 'о' Then

AllRecords[I].ElemDayOfWeek := '1'

Else

AllRecords[I].ElemDayOfWeek := '5'

End;

'В':

AllRecords[I].ElemDayOfWeek := '2';

'С':

Begin

If AllRecords[I].ElemDayOfWeek[2] = 'р' Then

AllRecords[I].ElemDayOfWeek := '3'

Else

AllRecords[I].ElemDayOfWeek := '6'

End;

'Ч':

AllRecords[I].ElemDayOfWeek := '4';

End;

If AllRecords[I].ElemTeacher = TeachName.Text Then

Begin

HaveTeacher := True;

Schedule.Cells[StrToInt(AllRecords[I].ElemDayOfWeek),

AllRecords[I].ElemNumOfLesson] := AllRecords[I].ElemSubject;

End;

End;

If HaveTeacher = False Then

Application.MessageBox('Данный учитель не найден', 'Предупреждение');

End;

Procedure TForm15.FormCreate(Sender: TObject);

Begin

Schedule.Cells[1, 0] := 'Понедельник';

Schedule.Cells[2, 0] := 'Вторник';

Schedule.Cells[3, 0] := 'Среда';

Schedule.Cells[4, 0] := 'Четверг';

Schedule.Cells[5, 0] := 'Пятница';

Schedule.Cells[6, 0] := 'Суббота';

Schedule.Cells[0, 0] := 'Пара';

Schedule.Cells[0, 1] := '1';

Schedule.Cells[0, 2] := '2';

Schedule.Cells[0, 3] := '3';

Schedule.Cells[0, 4] := '4';

Schedule.Cells[0, 5] := '5';

Schedule.Cells[0, 6] := '6';

End;

procedure TForm15.N1Click(Sender: TObject);

begin

Application.MessageBox('Для получения расписания в доступное поле введите имя преподавателя, затем нажмите кнопку заполнить', 'Инструкции');

end;

End.

**Код программы Java:**

package Lab4\_1;

import java.io.\*;

import java.util.Objects;

import java.util.Scanner;

public class Main {

public record MySchedule (String elemSubject, String elemTeacher, String elemDayOfWeek, byte elemNumOfLesson) {

}

final static String RECFROMFILE = "1";

final static String RECFROMCONSOLE = "2";

final static String SAVE = "3";

final static String SHOW = "4";

final static String DELETE = "5";

final static String ENDWORK = "6";

final static String HELP = "7";

final static String SHOWSCHEDULE = "8";

final static String MONDAY = "Понедельник";

final static String TUESDAY = "Вторник";

final static String WEDNESDAY = "Среда";

final static String THURSDAY = "Четверг";

final static String FRIDAY = "Пятница";

final static String SATURDAY = "Суббота";

static void callHelp() {

System.out.println("Для дополнения записей из файла необходимо в файле в первой строке ввести количество вводимых записей,\nзатем в каждой новой строке нужно вводить новый элемент добавляемой записи'+' (День недели(c большой буквы) -> Номер ' + 'пары(1..6) -> \nНазвание предмета(< 15 символов) -> ' + 'Имя преподавателя (< 15 символов).\n");

}

static void showMenu(){

System.out.println("Введите " + RECFROMFILE + " для добавления записи из файла, " + RECFROMCONSOLE

+ " для добавления записи из консоли" + '\n'

+ "Введите " + SAVE + " для сохранения текущих записей в файл" + '\n'

+ "Введите " + SHOW + ", чтобы увидеть текущие записи" + '\n'

+ "Введите " + DELETE + ", чтобы удалить запись" + '\n'

+ "Введите " + ENDWORK + ", чтобы остановить программу" + '\n'

+ "Введите " + SHOWSCHEDULE + ", чтобы открыть расписание преподавателя" + '\n'

+ "Введите " + HELP + ", чтобы получить помощь");

}

static MySchedule[] addRecordsFromFile(String path, MySchedule[] myRecords){

Scanner fin;

MySchedule[] myRecordsFromFile;

MySchedule myRecordFromFile;

String dayOfWeek;

String teacher;

String subject;

byte numOfLesson;

subject = "";

teacher = "";

dayOfWeek = "";

myRecordsFromFile = null;

boolean canReadFile;

int arraySize;

canReadFile = true;

arraySize = 0;

numOfLesson = 0;

fin = createFileStream(path);

try {

arraySize = Integer.parseInt(fin.nextLine());

}

catch (NumberFormatException formatException){

System.err.print("Не корректная информация в файле\n");

canReadFile = false;

}

if (canReadFile && (arraySize < 1 || arraySize > 100)) {

System.err.print("Заданное число записей вне диапазона\n");

canReadFile = false;

}

if (canReadFile) {

myRecordsFromFile = new MySchedule[arraySize];

for (int i = 0; i < arraySize; i++){

for (int j = 1; j <= 4; j++){

if (!fin.hasNext() && canReadFile){

System.err.print("Не все элементы введены\n");

canReadFile = false;

}

else if (canReadFile){

if (j == 1){

dayOfWeek = fin.nextLine();

if (!(Objects.equals(dayOfWeek, MONDAY)) && !(Objects.equals (dayOfWeek, TUESDAY)) && !(Objects.equals(dayOfWeek, WEDNESDAY)) && !(Objects.equals(dayOfWeek, THURSDAY)) &&(Objects.equals(dayOfWeek, FRIDAY)) && !(Objects.equals(dayOfWeek, SATURDAY))){

System.err.print("Не корректное заполнение файла\n");

canReadFile = false;

}

}

else if (j == 2){

try{

numOfLesson = Byte.parseByte(fin.nextLine());

}

catch(NumberFormatException formatException){

System.err.print("Не корректное заполнение файла\n");

canReadFile = false;

}

if (canReadFile && ((numOfLesson > 7) || (numOfLesson < 1))){

System.err.print("Не корректное заполнение файла\n");

canReadFile = false;

}

}

else if (j == 3){

subject = fin.nextLine();

if (subject.length() > 14){

System.err.print("Не корректное заполнение файла\n");

canReadFile = false;

}

}

else if (j == 4){

teacher = fin.nextLine();

if (teacher.length() > 14){

System.err.print("Не корректное заполнение файла\n");

canReadFile = false;

}

}

}

}

myRecordFromFile = new MySchedule(subject, teacher, dayOfWeek, numOfLesson);

if (canReadFile){

for (int k = 0; k < myRecords.length; k++) {

if ((Objects.equals(myRecordFromFile.elemDayOfWeek, myRecords[k].elemDayOfWeek))

&& (Objects.equals(myRecordFromFile.elemTeacher, myRecords[k].elemTeacher)) && (myRecordFromFile. elemNumOfLesson == myRecords[k].elemNumOfLesson)){

System.err.print("Один из элементов уже присутствует в записях\n");

canReadFile = false;

}

}

for (int k = 0; k < i; k++) {

if ((Objects.equals(myRecordFromFile.elemDayOfWeek, myRecordsFromFile[k].elemDayOfWeek))&& (Objects.equals(myRecordFromFile.elemTeacher, myRecordsFromFile[k].elemTeacher))

&& (myRecordFromFile.elemNumOfLesson == myRecordsFromFile[k].elemNumOfLesson)){

System.err.print("Один из элементов уже присутствует в записях\n");

canReadFile = false;

}

}

}

if (canReadFile)

myRecordsFromFile[i] = myRecordFromFile;

}

}

if (!canReadFile)

myRecordsFromFile = new MySchedule[0];

fin.close();

return myRecordsFromFile;

}

static MySchedule[] addRecords(MySchedule[] backUpForMyRecords, MySchedule[] newRecords){

MySchedule[] myRecords;

myRecords = new MySchedule[newRecords.length + backUpForMyRecords.length];

for (int i = 0; i < myRecords.length; i++){

if (i < backUpForMyRecords.length)

myRecords[i] = backUpForMyRecords[i];

else

myRecords[i] = newRecords[i - backUpForMyRecords.length];

}

return myRecords;

}

static MySchedule addRecordFromConsole(Scanner in){

MySchedule newSchedule;

boolean isIncorrect;

String dayOfWeek;

String teacher;

String subject;

byte numOfLesson;

numOfLesson = -1;

do {

isIncorrect = false;

System.out.println("Введите день недели (c большой буквы)");

dayOfWeek = in.nextLine();

if (!(Objects.equals(dayOfWeek, MONDAY)) && !(Objects.equals(dayOfWeek, TUESDAY)) && !(Objects.equals(dayOfWeek, WEDNESDAY)) && !(Objects.equals(dayOfWeek, THURSDAY)) && !(Objects.equals(dayOfWeek, FRIDAY)) && !(Objects.equals(dayOfWeek, SATURDAY))){

System.err.print("Не корректная запись\n");

isIncorrect = true;

}

} while(isIncorrect);

do {

isIncorrect = false;

System.out.println("Введите номер пары (1..6)");

try {

numOfLesson = Byte.parseByte(in.nextLine());

}

catch(NumberFormatException formatException){

System.err.print("Не корректная запись\n");

isIncorrect = true;

}

if (!isIncorrect && (numOfLesson < 1 || numOfLesson > 6)){

System.err.print("Значение не входит в диапазон\n");

isIncorrect = true;

}

} while(isIncorrect);

do {

isIncorrect = false;

System.out.println("Введите название предмета (< 15 символов)");

subject = in.nextLine();

if (subject.length() > 14){

System.err.print("Слишком длинная запись\n");

isIncorrect = true;

}

} while(isIncorrect);

do {

isIncorrect = false;

System.out.println("Введите имя преподователя (< 15 символов)");

teacher = in.nextLine();

if (teacher.length() > 14){

System.err.print("Слишком длинная запись\n");

isIncorrect = true;

}

} while(isIncorrect);

newSchedule = new MySchedule(subject, teacher, dayOfWeek, numOfLesson);

return newSchedule;

}

static void saveRecordsToFile(MySchedule[] myRecords, String path) {

Writer writer;

File file;

file = new File(path);

try {

writer = new FileWriter(file);

writer.append(String.valueOf(myRecords.length)).append("\n");

for (int k = 0; k < myRecords.length; k++) {

writer.append(myRecords[k].elemDayOfWeek).append("\n");

writer.append(String.valueOf(myRecords[k].elemNumOfLesson)).append("\n");

writer.append(myRecords[k].elemSubject).append("\n");

writer.append(myRecords[k].elemTeacher).append("\n");

}

writer.append('\n');

writer.close();

System.out.println("Запись в файл прошла успешно");

}

catch (IOException ioException){

System.out.println("Не удалось записать в файл");

}

}

static byte enterDeletedRecord(Scanner in, int restriction){

byte numOfRecord;

boolean isInCorrect;

numOfRecord = -1;

System.out.println("Введите номер удаляемой записи (1.." + (restriction) + ")");

do {

isInCorrect = false;

try{

numOfRecord = Byte.parseByte(in.nextLine());

}

catch(NumberFormatException formatException){

isInCorrect = true;

System.err.print("Введите число\n");

}

if (!isInCorrect && ((numOfRecord) > (restriction) || (numOfRecord) < 1)){

isInCorrect = true;

System.err.print("Введите число в допустимом диапазоне\n");

}

} while(isInCorrect);

numOfRecord--;

return numOfRecord;

}

static MySchedule[] deleteRecord(MySchedule[] myRecords,Scanner in){

byte numOfRecord;

MySchedule[] myNewRecords;

myNewRecords = new MySchedule[myRecords.length - 1];

numOfRecord = enterDeletedRecord(in, myRecords.length );

for (int i = 0; i < myRecords.length - 1; i++) {

if (i < numOfRecord) {

myNewRecords[i] = myRecords[i];

}

else {

myNewRecords[i] = myRecords[i + 1];

}

}

return myNewRecords;

}

static void showRecords(MySchedule[] myRecords){

System.out.println("Записи вида (День недели, номер пары, предмет, учитель");

for (int i = 0; i < myRecords.length; i++)

System.out.println("Запись " + i + ": " + myRecords[i].elemDayOfWeek + " " + myRecords[i].elemNumOfLesson + " " + myRecords[i].elemSubject + " " + myRecords[i].elemTeacher);

System.out.println();

}

static String[] createSchedule(MySchedule[] myRecords, String teacher){

String[] schedule;

schedule = new String[36];

for (int j = 0; j < 35; j++)

schedule[j] = "";

for (int i = 0; i < myRecords.length; i++) {

if (Objects.equals(myRecords[i].elemTeacher, teacher)) {

if (Objects.equals(myRecords[i].elemDayOfWeek, MONDAY)) {

schedule[myRecords[i].elemNumOfLesson - 1] = myRecords[i].elemSubject;

} else if (Objects.equals(myRecords[i].elemDayOfWeek, TUESDAY)) {

schedule[5 + myRecords[i].elemNumOfLesson] = myRecords[i].elemSubject;

} else if (Objects.equals(myRecords[i].elemDayOfWeek, WEDNESDAY)) {

schedule[11 + myRecords[i].elemNumOfLesson] = myRecords[i].elemSubject;

} else if (Objects.equals(myRecords[i].elemDayOfWeek, THURSDAY)) {

schedule[17 + myRecords[i].elemNumOfLesson] = myRecords[i].elemSubject;

} else if (Objects.equals(myRecords[i].elemDayOfWeek, FRIDAY)) {

schedule[23 + myRecords[i].elemNumOfLesson] = myRecords[i].elemSubject;

} else if (Objects.equals(myRecords[i].elemDayOfWeek, SATURDAY)) {

schedule[29 + myRecords[i].elemNumOfLesson] = myRecords[i].elemSubject;

}

}

}

return schedule;

}

static void showSchedule(MySchedule[] myRecords, String teacher){

String[] schedule;

schedule = new String[36];

schedule = createSchedule(myRecords, teacher);

System.out.println(MONDAY + ": ");

for (int i = 1; i < 6; i++){

if (!Objects.equals(schedule[i - 1], ""))

System.out.print(i + " пара: " + schedule[i - 1]);

}

System.out.println("\n" + TUESDAY + ": ");

for (int i = 1; i < 6; i++){

if (!Objects.equals(schedule[i + 5], ""))

System.out.print(i + " пара: " + schedule[i + 5]);

}

System.out.println("\n" + WEDNESDAY + ": ");

for (int i = 1; i < 6; i++){

if (!Objects.equals(schedule[i + 11], ""))

System.out.print(i + " пара: " + schedule[i + 11]);

}

System.out.println("\n" + THURSDAY + ": ");

for (int i = 1; i < 6; i++){

if (!Objects.equals(schedule[i + 17], ""))

System.out.print(i + " пара: " + schedule[i + 17]);

}

System.out.println("\n" + FRIDAY + ": ");

for (int i = 1; i < 6; i++){

if (!Objects.equals(schedule[i + 23], ""))

System.out.print(i + " пара: " + schedule[i + 23]);

}

System.out.println("\n" + SATURDAY + ": ");

for (int i = 1; i < 6; i++){

if (!Objects.equals(schedule[i + 29], ""))

System.out.print(i + " пара: " + schedule[i + 29]);

}

System.out.println();

}

static String enterTeacherName(Scanner in){

String teacher;

boolean isInCorrect;

do {

isInCorrect = false;

System.out.println("Введите имя преподователя (< 15 символов)");

teacher = in.nextLine();

if (teacher.length() > 14){

System.err.print("Слишком длинная запись\n");

isInCorrect = true;

}

} while(isInCorrect);

return teacher;

}

static void callMenu(Scanner in){

String path;

String instructions;

String teacher;

MySchedule[] myRecords;

MySchedule[] backUpForMyRecords;

MySchedule[] newRecords;

myRecords = new MySchedule[0];

instructions = "";

while (!Objects.equals(instructions, ENDWORK)){

showMenu();

instructions = in.nextLine();

if (Objects.equals(instructions, RECFROMFILE)) {

backUpForMyRecords = myRecords;

path = inputPathToFile(in);

newRecords = addRecordsFromFile(path, myRecords);

myRecords = addRecords(backUpForMyRecords, newRecords);

}

else if (Objects.equals(instructions, RECFROMCONSOLE)) {

backUpForMyRecords = myRecords;

newRecords = new MySchedule[1];

newRecords[0] = addRecordFromConsole(in);

myRecords = addRecords(backUpForMyRecords, newRecords);

}

else if (Objects.equals(instructions, SAVE)) {

if (myRecords.length > 0) {

path = inputPathToFile(in);

saveRecordsToFile(myRecords, path);

}

else

System.err.println("Нет записей");

}

else if (Objects.equals(instructions, SHOW)) {

if (myRecords.length > 0)

showRecords(myRecords);

else

System.err.println("Нет записей");

}

else if (Objects.equals(instructions, DELETE)){

if (myRecords.length > 0)

myRecords = deleteRecord(myRecords, in);

else

System.err.println("Нет записей");

}

else if (Objects.equals(instructions, HELP))

callHelp();

else if (Objects.equals(instructions, SHOWSCHEDULE)) {

if (myRecords.length > 0) {

teacher = enterTeacherName(in);

showSchedule(myRecords, teacher);

}

else

System.err.println("Нет записей");

}

else if (!Objects.equals(instructions, ENDWORK))

System.err.println("Неверная комманда");

}

}

static boolean isFileExists(String path){

boolean isNotCorrect;

isNotCorrect = false;

File inputFile;

inputFile = new File(path);

if (!inputFile.exists()) {

System.err.println("Файл не существует. Повторите ввод:");

isNotCorrect = true;

}

return (isNotCorrect);

}

static boolean isTypeCorrect(String path){

boolean isNotCorrect;

isNotCorrect = false;

if (!path.endsWith(".txt")) {

System.err.println("Файл должен быть формата .txt. Повторите ввод:");

isNotCorrect = true;

}

return (isNotCorrect);

}

static String inputPathToFile(Scanner in){

String path;

boolean isNotCorrect;

System.out.println("Укажите путь к файлу");

do{

isNotCorrect = false;

path = in.nextLine();

isNotCorrect = isFileExists(path);

if (!isNotCorrect)

isNotCorrect = isTypeCorrect(path);

} while(isNotCorrect);

return(path);

}

static Scanner createFileStream(String path){

Scanner fin;

File file;

fin = null;

file = new File(path);

try {

fin = new Scanner(file);

}

catch (FileNotFoundException e) {

System.out.println("Нельзя прочесть файл");

}

return fin;

}

public static void main(String[] args){

Scanner in;

in = new Scanner(System.in);

callMenu(in);

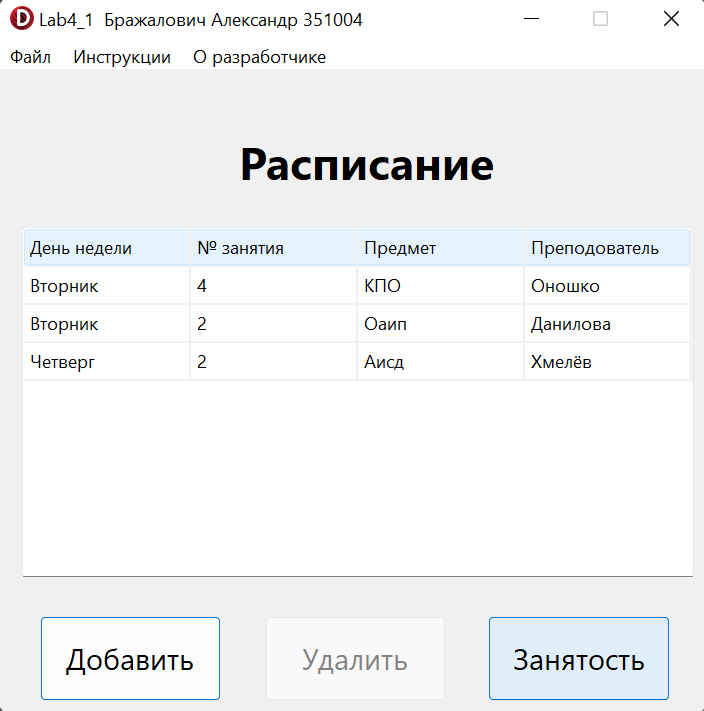
in.close();

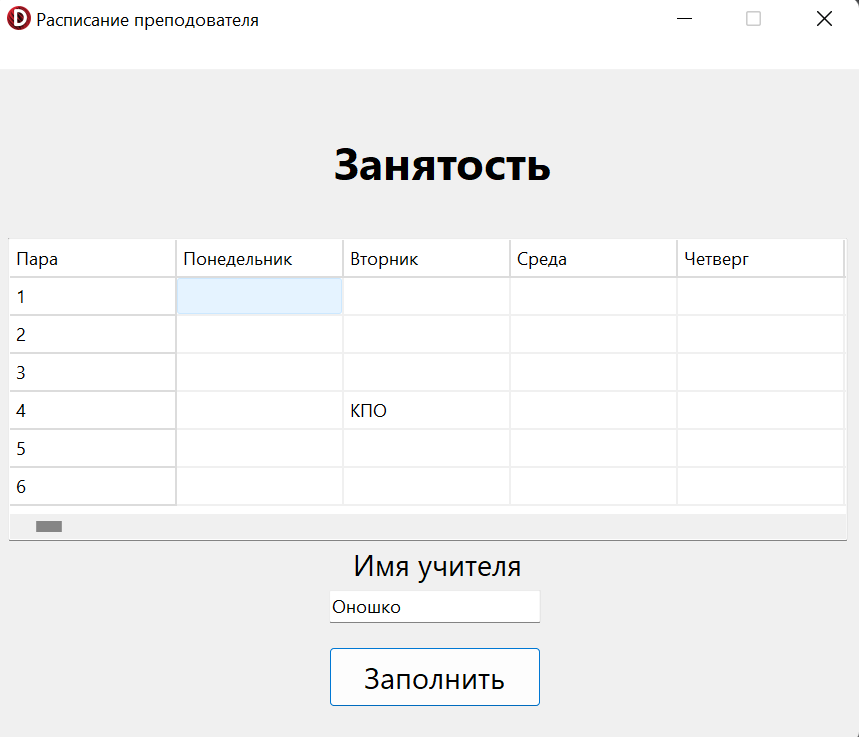
}

}

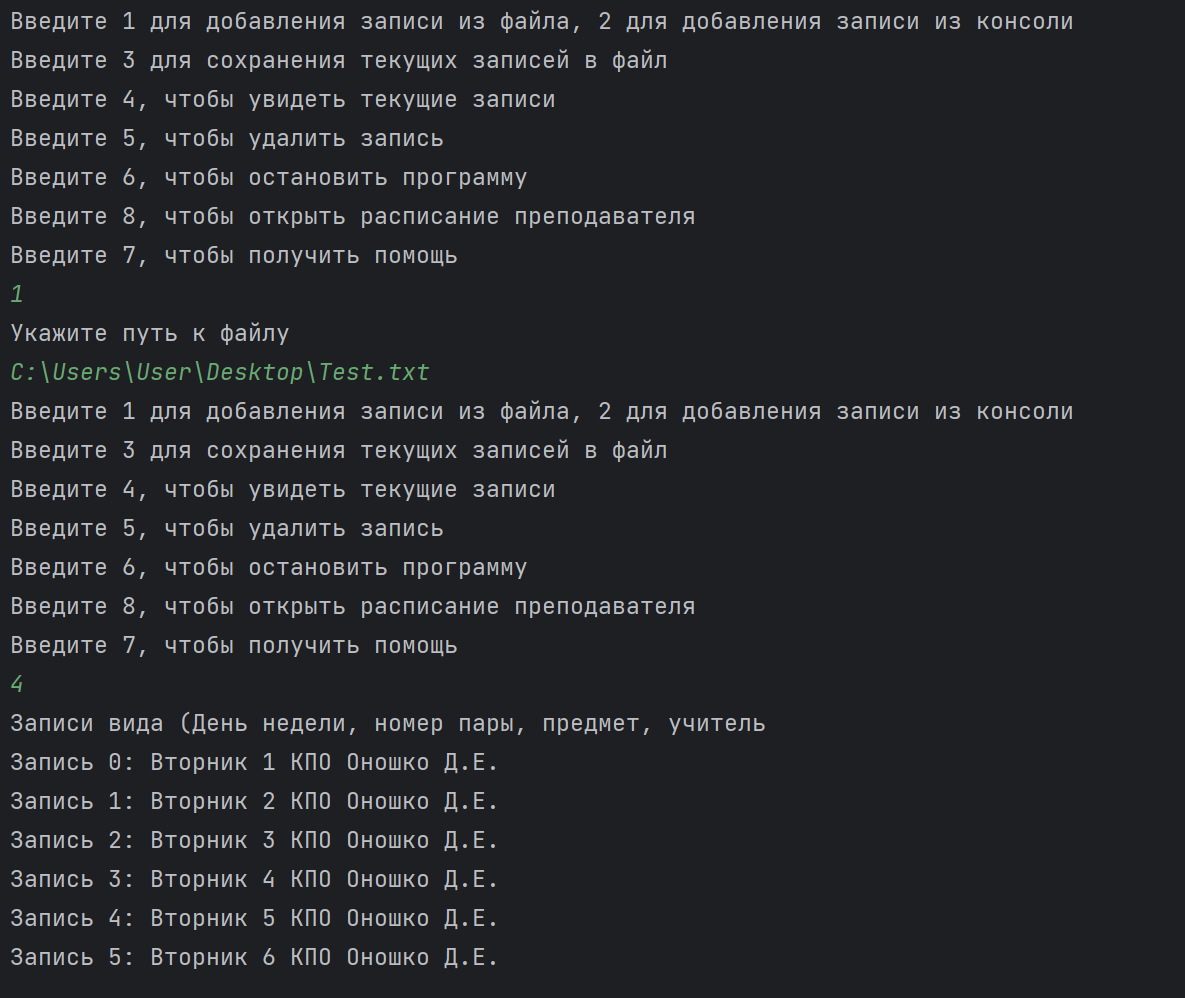
**Скриншоты:**

**Delphi:**



****

**Java:**



**Блок-схема:**

