**УО «Белорусский государственный университет информатики и**

**радиоэлектроники»**

**Кафедра ПОИТ**

**Отчет по лабораторной работе №4.1**

**по предмету**

**Основы Алгоритмизации и Программирования**

**Вариант 1**

**Выполнил**

**Андросов И.С.**

**Проверила**

**Данилова Г.В.**

Группа:

**8**51001

**Минск 2019**

**Задание**

**Записи**

Сведения о студенте колледжа:

код группы;

номер по журналу;

фамилия и инициалы;

дата рождения;

пол.

Вывести фамилии студентов отделения «Программирование» в порядке возрастания кодов групп, а внутри группы в порядке возрастания номера по журналу

**Код программы**

**(Delphi)**

**Students.pas**

**unit** Students;

**interface**

**uses**

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

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

**type**

TStudent = **Record**

GroupCode, Gender: String[6];

JornalNumber: String[2];

Surname: String[20];

DateOfBirth: String[10];

**end**;

TDataBase = **array of** TStudent;

TStartMenu = **class**(TForm)

Menu: TLabel;

StudentsList: TStringGrid;

AddBtn: TButton;

EditBtn: TButton;

DeleteBtn: TButton;

SortBtn: TButton;

SaveFile: TSaveDialog;

MainMenu: TMainMenu;

FIle1: TMenuItem;

Open: TMenuItem;

Save: TMenuItem;

N1: TMenuItem;

**Exit**: TMenuItem;

Help1: TMenuItem;

AboutTheProgram: TMenuItem;

AboutTheDeveloper: TMenuItem;

PopupMenu: TPopupMenu;

OpenFile: TOpenDialog;

**procedure** AddBtnClick(Sender: TObject);

**procedure** ExitClick(Sender: TObject);

**procedure** AboutTheProgramClick(Sender: TObject);

**procedure** AboutTheDeveloperClick(Sender: TObject);

**procedure** FormCloseQuery(Sender: TObject; **var** CanClose: Boolean);

**procedure** FormCreate(Sender: TObject);

**procedure** DeleteBtnClick(Sender: TObject);

**procedure** OpenClick(Sender: TObject);

**procedure** SaveClick(Sender: TObject);

**procedure** EditBtnClick(Sender: TObject);

**procedure** SortBtnClick(Sender: TObject);

**public**

DB: TDataBase;

**end**;

**var**

StartMenu: TStartMenu;

**implementation**

{$R \*.dfm}

**uses** ManageStudent, EditStudent;

**procedure** TStartMenu.AddBtnClick(Sender: TObject);

**begin**

Manage.Caption := 'Add student';

Manage.ShowModal;

**end**;

**procedure** TStartMenu.DeleteBtnClick(Sender: TObject);

**var**

ButtonSelected, i: ShortInt;

**begin**

**if** (StudentsList.Row <> 0) **then**

**begin**

ButtonSelected := MessageDlg('Are you sure you want to delete this record?',

mtConfirmation, [mbYes,mbNo], 0);

**if** ButtonSelected = mrYes **then**

**begin**

**for** i := (StudentsList.Row - 1) **to** (High(DB) - 1) **do**

**begin**

DB[i] := DB[i + 1];

**with** StudentsList **as** TStringGrid **do**

**begin**

Cells[0, i + 1] := DB[i].GroupCode;

Cells[1, i + 1] := DB[i].JornalNumber;

Cells[2, i + 1] := DB[i].Surname;

Cells[3, i + 1] := DB[i].DateOfBirth;

Cells[4, i + 1] := DB[i].Gender;

**end**;

**end**;

SetLength(DB, Length(DB) - 1);

StudentsList.RowCount := StudentsList.RowCount - 1;

**end**;

**if** StudentsList.RowCount = 1 **then**

**begin**

Save.Enabled := false;

EditBtn.Enabled := false;

DeleteBtn.Enabled := false;

SortBtn.Enabled := false;

**end**;

**end**;

**end**;

**procedure** TStartMenu.AboutTheProgramClick(Sender: TObject);

**begin**

MessageDlg('This program allows you to work with the university database.' + #13#10 + 'Button

''Sort'' orders the list by group code'+ #13#10 + 'and students by jornal number.' +

#13#10 + 'Ender datе of bith in dd.mm.yyyy format.', mtInformation, [mbOk], 0);

**end**;

**procedure** TStartMenu.AboutTheDeveloperClick(Sender: TObject);

**begin**

MessageDlg('This program is developed by Ilya Androsov' + #13 + 'BSUIR 2019', mtInformation,

[mbOk], 0);

**end**;

**procedure** TStartMenu.EditBtnClick(Sender: TObject);

**begin**

**if** (StudentsList.Row <> 0) **then**

**with** Manage **as** TManage **do**

**begin**

Caption := 'Edit student';

GC.Text := DB[StudentsList.Row - 1].GroupCode;

JN.Text := DB[StudentsList.Row - 1].JornalNumber;

SaI.Text := DB[StudentsList.Row - 1].Surname;

Day.Text := Copy(DB[StudentsList.Row - 1].DateOfBirth, 1, 2);

Month.Text := Copy(DB[StudentsList.Row - 1].DateOfBirth, 4, 2);

Year.Text := Copy(DB[StudentsList.Row - 1].DateOfBirth, 7, 4);

**if** DB[StudentsList.Row - 1].Gender = 'Female' **then**

Female.Checked := true

**else**

Male.Checked := true;

ShowModal;

**end**;

**end**;

**procedure** TStartMenu.ExitClick(Sender: TObject);

**begin**

Close;

**end**;

**procedure** TStartMenu.FormCloseQuery(Sender: TObject; **var** CanClose: Boolean);

**var**

ButtonSelected: Byte;

**begin**

ButtonSelected := MessageDlg('Are you sure you want to exit?', mtConfirmation, [mbYes,mbNo],

0);

**if** ButtonSelected <> mrYes **then**

CanClose := False;

**end**;

**procedure** TStartMenu.FormCreate(Sender: TObject);

**begin**

**with** StudentsList **as** TStringGrid **do**

**begin**

Cells[0, 0] := 'Group code';

Cells[1, 0] := 'Jornal number';

Cells[2, 0] := 'Surname and initials';

Cells[3, 0] := 'Date of birth';

Cells[4, 0] := 'Gender';

**end**;

**end**;

**procedure** TStartMenu.OpenClick(Sender: TObject);

**var**

InputFile: **File of** TStudent;

i, j: Byte;

Temp: SmallInt;

**begin**

**if** OpenFile.Execute **then**

**begin**

**try**

AssignFile(InputFile, OpenFile.FileName);

Reset(InputFile);

i := 0;

**repeat**

SetLength(DB, i + 1);

Read(InputFile, DB[i]);

Inc(i);

**until** EoF(InputFile);

StudentsList.RowCount := i + 1;

**with** StudentsList **as** TstringGrid **do**

**for** i := 0 **to** High(DB) **do**

**begin**

Cells[0, i + 1] := DB[i].GroupCode;

Cells[1, i + 1] := DB[i].JornalNumber;

Cells[2, i + 1] := DB[i].Surname;

Cells[3, i + 1] := DB[i].DateOfBirth;

Cells[4, i + 1] := DB[i].Gender;

**end**;

CloseFile(InputFile);

Save.Enabled := true;

EditBtn.Enabled := true;

DeleteBtn.Enabled := true;

**if** (StudentsList.RowCount > 1) **then**

SortBtn.Enabled := true

**else**

SortBtn.Enabled := false;

**except**

MessageDlg('Check entered data. Try again.', mtError, [mbRetry], 0);

CloseFile(InputFile);

**end**;

**end**;

**end**;

**procedure** TStartMenu.SaveClick(Sender: TObject);

**var**

OutputFile: **File of** TStudent;

MyFile: String;

ButtonSelected, i: Byte;

**begin**

**if** SaveFile.Execute **then**

**begin**

MyFile := SaveFile.FileName;

**if** FileExists(MyFile) **then**

**begin**

AssignFile(OutputFile, MyFile);

Rewrite(OutputFile);

**for** i := 0 **to** High(DB) **do**

Write(OutputFile, DB[i]);

CloseFile(outputFile);

**end**;

**end**;

**end**;

**function** OrderByJornal(MyArray: TDataBase): TDataBase;

**var**

ThatArray: TDataBase;

i, j, Left, Right, Size: Byte;

Temp: TStudent;

**begin**

Size := High(MyArray);

Left := Size;

Right := Size;

SetLength(ThatArray, Size \* 2 + 1);

ThatArray[Size] := MyArray[0];

**for** i := 1 **to** Size **do**

**begin**

Temp := MyArray[i];

**if** StrToInt(Temp.JornalNumber) >= StrToInt(MyArray[0].JornalNumber) **then**

**begin**

Inc(Right);

j := Right;

**while** StrToInt(Temp.JornalNumber) < StrToInt(ThatArray[j - 1].JornalNumber) **do**

**begin**

ThatArray[j] := ThatArray[j-1];

Dec(j);

**end**;

ThatArray[j] := Temp;

**end**

**else**

**begin**

Dec(Left);

j := Left;

**while** StrToInt(Temp.JornalNumber) > StrToInt(ThatArray[j + 1].JornalNumber) **do**

**begin**

ThatArray[j] := ThatArray[j + 1];

Inc(j);

**end**;

**end**;

ThatArray[j] := Temp;

**end**;

**for** j := 0 **to** Size **do**

MyArray[j] := ThatArray[j + Left];

OrderByJornal := MyArray;

**end**;

**function** OrderByGroup(MyArray: TDataBase): TDataBase;

**var**

ThatArray: TDataBase;

i, j, Left, Right, Size: Byte;

Temp: TStudent;

**begin**

Size := High(MyArray);

Left := Size;

Right := Size;

SetLength(ThatArray, Size \* 2 + 1);

ThatArray[Size] := MyArray[0];

**for** i := 1 **to** Size **do**

**begin**

Temp := MyArray[i];

**if** StrToInt(Temp.GroupCode) >= StrToInt(MyArray[0].GroupCode) **then**

**begin**

Inc(Right);

j := Right;

**while** StrToInt(Temp.GroupCode) < StrToInt(ThatArray[j - 1].GroupCode) **do**

**begin**

ThatArray[j] := ThatArray[j-1];

Dec(j);

**end**;

ThatArray[j] := Temp;

**end**

**else**

**begin**

Dec(Left);

j := Left;

**while** StrToInt(Temp.GroupCode) > StrToInt(ThatArray[j + 1].GroupCode) **do**

**begin**

ThatArray[j] := ThatArray[j + 1];

Inc(j);

**end**;

**end**;

ThatArray[j] := Temp;

**end**;

**for** j := 0 **to** Size **do**

MyArray[j] := ThatArray[j + Left];

OrderByGroup := MyArray;

**end**;

**procedure** TStartMenu.SortBtnClick(Sender: TObject);

**var**

i, j, k: Byte;

TempArray: TDataBase;

**begin**

OrderByGroup(DB);

j := 0;

i := 0;

**with** StudentsList **as** TstringGrid **do**

**begin**

**repeat**

k := 0;

**repeat**

SetLength(TempArray, k + 1);

TempArray[k] := DB[i];

Inc(k);

Inc(I);

**until** (DB[i - 1].GroupCode <> DB[i].GroupCode);

OrderByJornal(TempArray);

k := 0;

**repeat**

DB[j] := TempArray[k];

Inc(j);

Inc(k);

**until** j > i - 1;

**until** i > High(DB);

**for** i := 0 **to** High(DB) **do**

**begin**

Cells[0, i + 1] := DB[i].GroupCode;

Cells[1, i + 1] := DB[i].JornalNumber;

Cells[2, i + 1] := DB[i].Surname;

Cells[3, i + 1] := DB[i].DateOfBirth;

Cells[4, i + 1] := DB[i].Gender;

**end**;

**end**;

**end**;

**end**.

**ManageStudent.pas**

**unit** ManageStudent;

**interface**

**uses**

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

Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Samples.Spin,

Vcl.DBCtrls, Vcl.Grids;

**type**

TManage = **class**(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

CancelBtn: TButton;

SubmitBtn: TButton;

Label6: TLabel;

GC: TEdit;

JN: TEdit;

SaI: TEdit;

Month: TEdit;

Day: TEdit;

Year: TEdit;

Label7: TLabel;

Label8: TLabel;

Female: TCheckBox;

Male: TCheckBox;

**procedure** CancelBtnClick(Sender: TObject);

**procedure** SubmitBtnClick(Sender: TObject);

**procedure** GCKeyPress(Sender: TObject; **var** Key: Char);

**procedure** OnChange(Sender: TObject);

**procedure** JNKeyPress(Sender: TObject; **var** Key: Char);

**procedure** SaIKeyPress(Sender: TObject; **var** Key: Char);

**procedure** DayKeyPress(Sender: TObject; **var** Key: Char);

**procedure** DayKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**procedure** MonthKeyPress(Sender: TObject; **var** Key: Char);

**procedure** MonthKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**procedure** YearKeyPress(Sender: TObject; **var** Key: Char);

**procedure** YearKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**procedure** FemaleClick(Sender: TObject);

**procedure** MaleClick(Sender: TObject);

**procedure** AddStudent(Sender: TObject);

**procedure** EditStudent(Sender: TObject);

**procedure** JNKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**end**;

**var**

Manage: TManage;

**implementation**

{$R \*.dfm}

**uses** Students;

**procedure** TManage.CancelBtnClick(Sender: TObject);

**begin**

GC.Clear;

JN.Clear;

SaI.Clear;

Day.Clear;

Month.Clear;

Year.Clear;

Female.Checked := false;

Male.Checked := false;

Close;

**end**;

**procedure** TManage.OnChange(Sender: TObject);

**begin**

**if** (Length(GC.Text) > 0)

**and** (Length(JN.Text) > 0)

**and** (Length(SaI.Text) > 0)

**and** (Length(Day.Text) = 2)

**and** (Length(Month.Text) = 2)

**and** (Length(Year.Text) = 4)

**and** ((Male.Checked) **or** (Female.Checked))

**then**

SubmitBtn.Enabled := true

**else**

SubmitBtn.Enabled := false;

**end**;

**procedure** TManage.FemaleClick(Sender: TObject);

**begin**

Male.Checked := false;

**end**;

**procedure** TManage.MaleClick(Sender: TObject);

**begin**

Female.Checked := false;

**end**;

**procedure** TManage.DayKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **set of** Char;

**begin**

Numerals := ['0'..'9', #8];

**with** Day **as** TEdit **do**

**begin**

**if not**(Key **in** Numerals) **then**

Key := #0;

**if** (Length(Text) = 2) **and** (Key <> #8)**then**

Key := #0;

**end**;

**end**;

**procedure** TManage.DayKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**begin**

**with** Day **as** Tedit **do**

**begin**

**if** (Length(Text) = 2) **and** ((StrToInt(Text) > 31) **or** (StrToInt(Text) = 0)) **then**

Text := '';

**end**;

**end**;

**procedure** TManage.GCKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **set of** Char;

**begin**

Numerals := ['0'..'9', #8];

**with** GC **as** TEdit **do**

**begin**

**if not**(Key **in** Numerals) **then**

Key := #0;

**if** (Length(Text) = 6) **and** (Key <> #8)**then**

Key := #0

**end**;

**end**;

**procedure** TManage.JNKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **set of** Char;

**begin**

Numerals := ['0'..'9', #8];

**with** JN **as** TEdit **do**

**begin**

**if not**(Key **in** Numerals) **then**

Key := #0;

**if** (Length(Text) = 2) **and** (Key <> #8) **then**

Key := #0;

**if** (Length(Text) = 0) **and** (Key = '0') **then**

Key := #0;

**end**;

**end**;

**procedure** TManage.JNKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**begin**

**with** JN **as** Tedit **do**

**if** (Text = '0') **then**

Text := '';

**end**;

**procedure** TManage.MonthKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **set of** Char;

**begin**

Numerals := ['0'..'9', #8];

**with** Month **as** TEdit **do**

**begin**

**if not**(Key **in** Numerals) **then**

Key := #0;

**if** (Length(Text) = 2) **and** (Key <> #8)**then**

Key := #0;

**end**;

**end**;

**procedure** TManage.MonthKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**begin**

**with** Month **as** Tedit **do**

**begin**

**if** (Length(Text) = 2) **and** ((StrToInt(Text) > 12) **or** (StrToInt(Text) = 0)) **then**

Text := '';

**end**;

**end**;

**procedure** TManage.SaIKeyPress(Sender: TObject; **var** Key: Char);

**var**

Letters: **set of** Char;

**begin**

Letters := ['A'..'z', '.', '-', ' ', #8];

**with** SaI **as** TEdit **do**

**begin**

**if not**(Key **in** Letters) **then**

Key := #0;

**if** (Length(Text) = 20) **and** (Key <> #8) **then**

Key := #0

**end**;

**end**;

**procedure** TManage.EditStudent(Sender: TObject);

**begin**

**with** StartMenu **as** TStartMenu **do**

**begin**

DB[StudentsList.Row - 1].GroupCode := GC.Text;

DB[StudentsList.Row - 1].JornalNumber := JN.Text;

DB[StudentsList.Row - 1].Surname := SaI.Text;

DB[StudentsList.Row - 1].DateOfBirth := Day.Text + '.' + Month.Text + '.' + Year.Text;

**if** Female.Checked **then**

DB[StudentsList.Row - 1].Gender := 'Female'

**else**

DB[StudentsList.Row - 1].Gender := 'Male';

**with** StudentsList **as** TStringGrid **do**

**begin**

Cells[0, Row] := DB[Row - 1].GroupCode;

Cells[1, Row] := DB[Row - 1].JornalNumber;

Cells[2, Row] := DB[Row - 1].Surname;

Cells[3, Row] := DB[Row - 1].DateOfBirth;

Cells[4, Row] := DB[Row - 1].Gender;

**end**;

**end**;

**end**;

**procedure** TManage.AddStudent(Sender: TObject);

**begin**

**with** StartMenu **as** TStartMenu **do**

**begin**

SetLength(DB, Length(DB) + 1);

DB[High(DB)].GroupCode := GC.Text;

DB[High(DB)].JornalNumber := JN.Text;

DB[High(DB)].Surname := SaI.Text;

DB[High(DB)].DateOfBirth := Day.Text + '.' + Month.Text + '.' + Year.Text;

**if** Female.Checked **then**

DB[High(DB)].Gender := 'Female'

**else**

DB[High(DB)].Gender := 'Male';

**with** StudentsList **as** TStringGrid **do**

**begin**

RowCount := RowCount + 1;

Cells[0, Length(DB)] := DB[High(DB)].GroupCode;

Cells[1, Length(DB)] := DB[High(DB)].JornalNumber;

Cells[2, Length(DB)] := DB[High(DB)].Surname;

Cells[3, Length(DB)] := DB[High(DB)].DateOfBirth;

Cells[4, Length(DB)] := DB[High(DB)].Gender;

**end**;

**end**;

StartMenu.Save.Enabled := true;

**end**;

**procedure** TManage.SubmitBtnClick(Sender: TObject);

**begin**

**if** Caption = 'Add student' **then**

AddStudent(Sender)

**else**

EditStudent(Sender);

GC.Clear;

JN.Clear;

SaI.Clear;

Day.Clear;

Month.Clear;

Year.Clear;

Female.Checked := false;

Male.Checked := false;

**with** StartMenu **as** TStartMenu **do**

**begin**

Save.Enabled := true;

EditBtn.Enabled := true;

DeleteBtn.Enabled := true;

**if** (StudentsList.RowCount > 2) **then**

SortBtn.Enabled := true

**else**

SortBtn.Enabled := false;

**end**;

Close;

**end**;

**procedure** TManage.YearKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **set of** Char;

**begin**

Numerals := ['0'..'9', #8];

**with** Year **as** TEdit **do**

**begin**

**if not**(Key **in** Numerals) **then**

Key := #0;

**if** (Length(Text) = 4) **and** (Key <> #8)**then**

Key := #0;

**end**;

**end**;

**procedure** TManage.YearKeyUp(Sender: TObject; **var** Key: Word; Shift: TShiftState);

**begin**

**with** Year **as** Tedit **do**

**if** (Length(Text) = 4) **and** ((StrToInt(Text) > 2013) **or** (StrToInt(Text) < 1919)) **then**

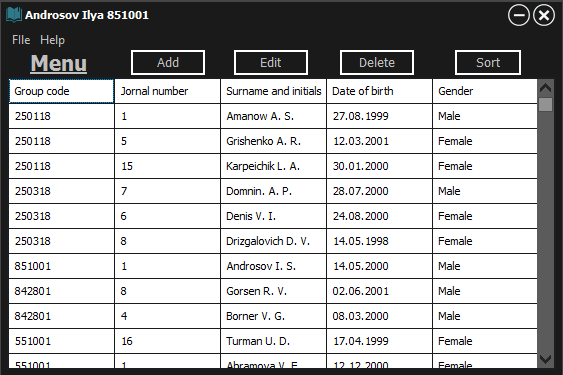
Text := '';

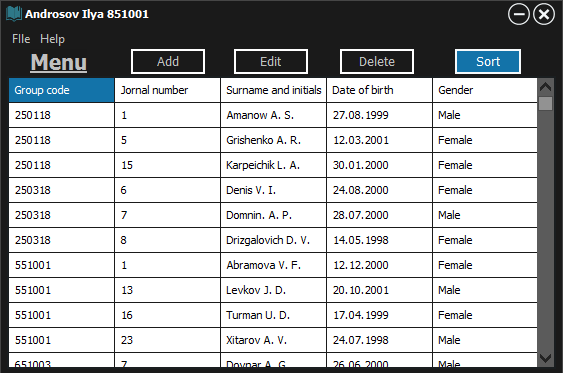
**end**;

**end**.

**Скриншоты**

**(Delphi)**

****

****

**Схема алгоритма**







