**УО «Белорусский государственный университет информатики и**

**радиоэлектроники»**

**Кафедра ПОИТ**

**Отчет по лабораторной работе №5.2**

**по предмету**

**Основы Алгоритмизации и Программирования**

**Вариант 1**

**Выполнил**

**Андросов И.С.**

**Проверила**

**Данилова Г.В.**

Группа:

**8**51001

**Минск 2019**

**Задание**

Даны n точек, заданные своими координатами. Найти прямую, через которую проходит максимальное количество точек. Визуализировать.

**Код программы**

**(Delphi)**

**Unit** Unit1;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.**Var**iants, System.Classes, Vcl.Graphics,

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

Vcl.ExtCtrls;

type

TLab\_5\_2 = class(TForm)

Label1: TLabel;

Num: TEdit;

StringGrid: TStringGrid;

MainMenu1: TMainMenu;

PopupMenu1: TPopupMenu;

SaveFile: TSaveDialog;

OpenFile: TOpendialog;

File1: TMenuItem;

Open: TMenuItem;

Save: TMenuItem;

N1: TMenuItem;

Exit: TMenuItem;

Help: TMenuItem;

Abouttheprogram1: TMenuItem;

Aboutthedeveloper1: TMenuItem;

Label2: TLabel;

Create: TButton;

Result: TLabel;

TryAgain: TButton;

Calculate: TButton;

Image1: TImage;

**procedure** Aboutthedeveloper1Click(Sender: TObject);

**procedure** Abouttheprogram1Click(Sender: TObject);

**procedure** GetColName(GridSize: Byte);

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

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

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

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

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

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

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

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

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

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

**procedure** Grafic(A, B, C, D: Integer; Sender: TObject);

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**var**

Lab\_5\_2: TLab\_5\_2;

**implementation**

{$R \*.dfm}

**procedure** TLab\_5\_2.Aboutthedeveloper1Click(Sender: TObject);

**begin**

MessageDlg('This program is developed by Androsov Ilya, group

851001', mtInformation, [mbOk], 0);

**end**;

**procedure** TLab\_5\_2.Abouttheprogram1Click(Sender: TObject);

**begin**

MessageDlg('This program find the equation of the line which belongs the

biggest number of the points', mtInformation, [mbOk], 0);

**end**;

**procedure** TLab\_5\_2.Grafic(A, B, C, D: Integer; S**end**er: TObject);

**const**

x1 = -30;

x2 = 30;

mash = 10;

**var**

x0, y0, i, j, k, Numb: integer;

x, y: real;

**begin**

Numb := StrToInt(Num.Text);

**with** Image1 **do**

**begin**

x0 := width div 2;

y0 := height div 2;

**with** Canvas do

**begin**

Pen.Color := clBlack;

Pen.Width := 1;

MoveTo(0,y0);

Lineto(Width, y0);

MoveTo(x0,0);

LineTo(x0,Height);

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

**for** j := -1 **to** 1 **do**

**for** k := -1 **to** 1 **do**

Pixels[x0 + StrToInt(StringGrid.Cells[i,1])\* **mash** + j, y0 –

StrToInt(StringGrid.Cells[i,2])\* mash + k] := clBlue;

Pen.Color := clRed;

Pen.Width := 2;

**if** C - A <> 0 **then**

**begin**

x := x1;

y := (x \* (D - B) + (D \*(C - A) - C \* (D - B)));

MoveTo(x0 + Trunc(x\*mash),y0- Trunc(y\*mash)\*(C - A));

x := x2;

y := (x \* (D - B) + (D \*(C - A) - C \* (D - B)));

LineTo(x0 + Trunc(x\*mash),y0 - Trunc(y \* mash)\*(C - A));

**end**

**else**

**begin**

x := A;

MoveTo(x0 + Trunc(x\*mash),0);

LineTo(x0 + Trunc(x\*mash),Height);

**end**;

**end**;

**end**;

**end**;

**procedure** TLab\_5\_2.CalculateClick(Sender: TObject);

**var**

Empty: Boolean;

Arr : array [1.. 256] of integer;

i, j, k, Number, h, m, max: Integer;

**begin**

Number := StrToInt(Num.Text);

Empty := False;

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

**begin**

**if** StringGrid.Cells[i, 1] = '' **then**

**begin**

Empty := True;

StringGrid.Cells[i, 1] := '0';

**end**;

**if** StringGrid.Cells[i, 2] = '' **then**

**begin**

Empty := True;

StringGrid.Cells[i, 2] := '0';

**end**;

**end**;

**if** (Empty) **then**

MessageDlg('In the sequence there are empty fields, the will be

replaced by zero.', mtInformation, [mbOk], 0);

m := 1;

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

**for** j := 1 + i **to** Number **do**

**begin**

h := 2;

**for** k := 1 + j **to** Number **do**

**begin**

**if** (StrToInt(StringGrid.Cells[k,1]) –

StrToInt(StringGrid.Cells[j,1])) \*

(StrToInt(StringGrid.Cells[j,2])-

StrToInt(StringGrid.Cells[i,2])) =

(StrToInt(StringGrid.Cells[k,2]) –

StrToInt(StringGrid.Cells[j,2])) \*

(StrToInt(StringGrid.Cells[j,1])-

StrToInt(StringGrid.Cells[i,1])) **then**

inc(h);

**end**;

Arr[m] := h;

inc(m);

**end**;

max := Arr[1];

j := 1;

**for** i := 2 **to** m **do**

**if** max < Arr[i] **then**

**begin**

j := i;

max := Arr[i];

**end**;

i := 1;

**while** j <> 1 **do**

**begin**

j:= j - number + i;

inc(i);

**end**;

j := StrToInt(StringGrid.Cells[i,1]);

k := StrToInt(StringGrid.Cells[i,2]);

h := StrToInt(StringGrid.Cells[i+1,1]);

m := StrToInt(StringGrid.Cells[i + 1,2]);

**if** ((m \*(h - j) - h \* (m - k)) < 0) **then**

Result.Caption := IntToStr(h - j) + 'y = ' + IntToStr(m - k) + 'x' +

IntToStr(m \*(h - j) - h \* (m - k))

**else**

Result.Caption := IntToStr(h - j) + 'y = ' + IntToStr(m - k) + 'x + ' +

IntToStr(m \*(h - j) - h \* (m - k));

TryAgain.Visible := True;

Calculate.Enabled := False;

Save.Enabled := True;

Grafic(j, k, h, m, Sender);

**end**;

**procedure** TLab\_5\_2.CreateClick(Sender: TObject);

**var**

GridSize: Byte;

**begin**

**if** (Length(Num.Text) = 0) **then**

**begin**

MessageDlg('Incorrect input, size of the array shoud be more **then** 0',

mtError, [mbOk], 0);

Num.Clear;

**end**

**else**

**begin**

GridSize := StrToInt(Num.Text);

StringGrid.ColCount := GridSize + 1;

StringGrid.Enabled := True;

Num.Enabled := False;

Calculate.Enabled := True;

Save.Enabled := False;

Create.Enabled := False;

GetColName(GridSize);

**end**;

**end**;

**procedure** TLab\_5\_2.ExitClick(Sender: TObject);

**begin**

Close;

**end**;

**procedure** TLab\_5\_2.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**;

**function** CheckFileName(MyFile: String): String;

**var**

i: Byte;

IsCorrect: Boolean;

**begin**

IsCorrect := False;

i := 1;

**while** **not** IsCorrect **and** (i <= Length(MyFile)) **do**

**begin**

**if** MyFile[i] = '.' **then**

IsCorrect := True;

Inc(i);

**end**;

**if** **not** IsCorrect **then**

MyFile := MyFile + '.txt';

CheckFileName := MyFile;

**end**;

**procedure** TLab\_5\_2.GetColName(GridSize: Byte);

**var**

I, J: Byte;

**begin**

StringGrid.Cells[0,1] := 'X';

StringGrid.Cells[0,2] := 'Y';

**for** I := 1 **to** GridSize **do**

**for** J := 1 **to** GridSize **do**

**begin**

StringGrid.Cells[I, 0] := IntToStr(i);

StringGrid.Cells[J, I] := '0';

**end**;

**end**;

**procedure** TLab\_5\_2.NumChange(Sender: TObject);

**begin**

Save.Enabled := False;

**if** (Length(Num.Text) > 0) **then**

Create.Enabled := True

**else**

Create.Enabled := False;

**end**;

**procedure** TLab\_5\_2.NumKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: set of Char;

**begin**

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

**with** Sender **as** TEdit **do**

**begin**

**if** (Key = #13) **and** (Length(Text) = 0) **then**

Key := #0;

**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** TLab\_5\_2.OpenClick(Sender: TObject);

**var**

InputFile: TextFile;

Temp: Double;

i, j: Byte;

**begin**

Num.Clear;

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

**begin**

**try**

AssignFile(InputFile, OpenFile.FileName);

Reset(InputFile);

**if** SeekEof(InputFile) **then**

**begin**

MessageDlg('This file is empty. Try again.', mtError, [mbRetry],

0);

CloseFile(InputFile);

**end**

**else**

**begin**

Readln(InputFile, Temp);

Num.Text := FloatToStr(Temp);

CreateClick(S**end**er);

i := 1;

**while** (**not** Eof(InputFile)) **do**

**begin**

j := 1;

**while** (**not** Eoln(InputFile)) **do**

**begin**

Read(InputFile, Temp);

StringGrid.Cells[j,i] := FloatToStr(Temp);

inc(j);

**end**;

Readln(InputFile);

inc(i);

**end**;

**if** j > 3 **then**

MessageDlg('Check entered data. Try again.', mtError,

[mbRetry], 0);

CloseFile(InputFile);

**end**;

**except**

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

Num.Clear;

CloseFile(InputFile);

**end**;

**end**;

**end**;

**procedure** TLab\_5\_2.SaveClick(Sender: TObject);

**var**

OutputFile: TextFile;

MyFile: String;

ButtonSelected, i, j: Byte;

**begin**

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

**begin**

MyFile := SaveFile.FileName;

MyFile := CheckFileName(MyFile);

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

**begin**

ButtonSelected := MessageDlg('Do you want to rewrite the file?',

mtConfirmation, [mbYes,mbNo], 0);

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

**begin**

AssignFile(OutputFile, MyFile);

Rewrite(OutputFile);

Writeln(Result.Caption);

CloseFile(OutputFile);

**end**

**else**

**begin**

**As**signFile(OutputFile, MyFile);

Append(OutputFile);

Writeln(Result.Caption);

CloseFile(OutputFile);

**end**;

**end**

**else**

**begin**

AssignFile(OutputFile, MyFile);

Rewrite(OutputFile);

Writeln(Result.Caption);

CloseFile(OutputFile);

**end**;

**end**;

**end**;

**procedure** TLab\_5\_2.StringGridKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: set of Char;

**begin**

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

**with** Sender **as** TStringgrid do

**begin**

**if** (Key = #13) **and** (Length(Cells[Col,Row]) = 0) **then**

Key := #0;

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

Key := #0;

**if** (Length(Cells[Col,Row]) > 0) **and** (Key = '-') **then**

Key := #0;

**if** (Length(Cells[Col,Row]) = 2) **and** (Key <> #8) **then**

Key := #0;

**if** (Cells[col,Row] <> '-') **and** (Length(Cells[Col,Row]) = 1) **and** (Key <> #8)**then**

Key := #0

**end**;

**end**;

**procedure** TLab\_5\_2.TryAgainClick(S**end**er: TObject);

**var**

i, j: Byte;

**begin**

**for** i := 0 **to** StrToInt(Num.Text) **do**

**for** j := 0 **to** StrToInt(Num.Text) **do**

StringGrid.Cells[i,j] := '';

Num.Clear;

Num.Enabled := True;

Result.Caption := '';

StringGrid.Enabled := False;

Save.Enabled := False;

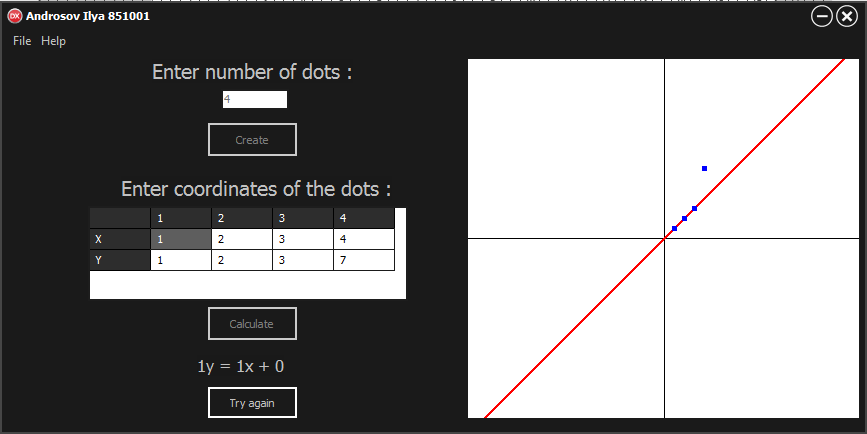
TryAgain.Visible := False;

Image1.Picture := nil;

**end**;

**end**.

**Скриншоты**



**Блок-схема**







