**УО «Белорусский государственный университет информатики и радиоэлектроники»**

**Кафедра ПОИТ**

**Отчёт по лабораторной работе №7.2**

**По предмету**

**Основы алгоритмизации и программирования**

**Вариант 6 (9)**

**Выполнил:**

**Гладкий М.Г.**

**Проверила:**

**Данилова Г.В.**

**Группа 851001**

**Минск 2019**

**Задание:**

Преобразовать матрицу инциденций в списки инцидентности.

**Delphi 10:**

**Main.pas**

**unit** Main;

**interface**

**uses**

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

System.Classes, Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs,

Vcl.Menus, Vcl.Grids, Vcl.StdCtrls;

**type**

TMainF = **class**(TForm)

SG: TStringGrid;

MainMenu: TMainMenu;

OpenFile: TOpenDialog;

PopupMenu: TPopupMenu;

FileMenu: TMenuItem;

Open: TMenuItem;

Help: TMenuItem;

Info: TMenuItem;

Developer: TMenuItem;

TransformBtn: TButton;

SetSizeBtn: TButton;

VerticesLbl: TLabel;

VertEdit: TEdit;

EdgesLbl: TLabel;

EdgeEdit: TEdit;

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

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

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

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

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

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

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

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

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

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

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

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

**end**;

**var**

MainF: TMainF;

**implementation**

{$R \*.dfm}

**uses** List;

**procedure** TMainF.DeveloperClick(Sender: TObject);

**begin**

MessageDlg('Developer: Gladkiy Maksim, gp.851001', mtInformation,

[mbOk], 0);

**end**;

**procedure** TMainF.InfoClick(Sender: TObject);

**begin**

MessageDlg('This program converts an incidence matrix to an incidence

list.' + #13#10 + 'Vertices[2..9].' + #13#10 + 'Edges[1..9].' +

#13#10 + 'In table cells use 0 or 1.', mtInformation, [mbOk], 0);

**end**;

**procedure** TMainF.TransformBtnClick(Sender: TObject);

**var**

i, j, k, l: ShortInt;

IsEmpty: Boolean;

**begin**

IsEmpty := false;

**for** j := 1 **to** SG.RowCount - 1 **do**

**for** i := 1 **to** SG.ColCount - 1 **do**

**if** (SG.Cells[i, j] = '') **then**

**begin**

IsEmpty := true;

SG.Cells[i, j] := '0';

**end**;

**if** IsEmpty **then**

MessageDlg('Empty cells were filled with zeros.',

mtConfirmation, [mbOk], 0);

**for** j := 1 **to** SG.RowCount **do**

**begin**

l := 1;

**for** i := 1 **to** SG.ColCount **do**

**if** (SG.Cells[i, j] = '1') **then**

**for** k := 1 **to** SG.RowCount **do**

**begin**

**if** (SG.Cells[i, k] = '1')**and**(k <> j) **then**

**begin**

ListF.ListSG.Cells[l, j - 1] := '=>';

ListF.ListSG.Cells[l + 1, j - 1] := IntToStr(k);

inc(l, 2);

**end**;

**end**;

ListF.ListSG.Cells[l, j - 1] := '=>';

ListF.ListSG.Cells[l + 1, j - 1] := 'nil';

**end**;

ListF.ShowModal;

**end**;

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

**var**

i: ShortInt;

**begin**

**for** i := 1 **to** 9 **do**

**begin**

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

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

**end**;

SG.Cells[0, 0] := 'H';

SG.FixedCols := 1;

SG.FixedRows := 1;

**end**;

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

**var**

InputFile: TextFile;

i, j, Temp: ShortInt;

IsCorrect: Boolean;

**begin**

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

**begin**

VertEdit.Text := '';

EdgeEdit.Text := '';

**try**

AssignFile(InputFile, OpenFile.FileName);

Reset(InputFile);

**if** EoF(InputFile) **then**

**begin**

MessageDlg('This file is empty. Try again.',

mtError, [mbRetry], 0);

CloseFile(InputFile);

**end**

**else**

**begin**

**repeat**

Read(InputFile, Temp);

**if** (Temp <> 0) **and** (Temp <> 1) **then**

IsCorrect := false

**until** EoF(InputFile) **or not**(IsCorrect);

**if** IsCorrect **then**

**begin**

Reset(InputFile);

j := 0;

**repeat**

i := 0;

**repeat**

Read(InputFile, Temp);

Inc(i);

**until** EoLn(InputFile);

inc(j);

**until** EoF(InputFile);

VertEdit.Text := IntToStr(j);

EdgeEdit.Text := IntToStr(i);

SetSizeBtn.Click;

Reset(InputFile);

**for** j := 1 **to** SG.RowCount - 1 **do**

**for** i := 1 **to** SG.ColCount - 1 **do**

**begin**

Read(InputFile, Temp);

SG.Cells[i, j] := IntToStr(Temp);

**end**;

TransformBtn.Click;

CloseFile(InputFile);

**end**

**else**

**begin**

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

mtError, [mbRetry], 0);

CloseFile(InputFile);

**end**;

**end**;

**except**

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

mtError, [mbRetry], 0);

CloseFile(InputFile);

**end**;

**end**;

**end**;

**procedure** TMainF.SetSizeBtnClick(Sender: TObject);

**begin**

SG.ColCount := StrToInt(EdgeEdit.Text) + 1;

SG.RowCount := StrToInt(VertEdit.Text) + 1;

ListF.ListSG.ColCount := StrToInt(EdgeEdit.Text) \* 2 + 3;

ListF.ListSG.RowCount := StrToInt(VertEdit.Text);

ListF.ClientHeight := 33 \* ListF.ListSG.RowCount;

ListF.ClientWidth := 33 \* ListF.ListSG.ColCount;

SetSizeBtn.Enabled := false;

SG.Enabled := true;

TransformBtn.Enabled := true;

**end**;

**procedure** TMainF.SGKeyPress(Sender: TObject; **var** Key: Char);

**begin**

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

Key := #0;

**if** (Key <> '0') **and** (Key <> '1') **and** (Key <> #8) **then**

Key := #0;

**end**;

**procedure** TMainF.VertEditChange(Sender: TObject);

**var**

i, j: ShortInt;

**begin**

TransformBtn.Enabled := false;

SG.Enabled := false;

**for** j := 1 **to** SG.RowCount - 1 **do**

**for** i := 1 **to** SG.ColCount - 1 **do**

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

**if** (Length(VertEdit.Text) = 1)**and**(Length(EdgeEdit.Text) = 1) **then**

SetSizeBtn.Enabled := true

**else**

SetSizeBtn.Enabled := false;

**end**;

**procedure** TMainF.EdgeEditChange(Sender: TObject);

**var**

i, j: ShortInt;

**begin**

TransformBtn.Enabled := false;

SG.Enabled := false;

**for** j := 1 **to** SG.RowCount - 1 **do**

**for** i := 1 **to** SG.ColCount - 1 **do**

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

**if** (Length(VertEdit.Text) = 1)**and**(Length(EdgeEdit.Text) = 1) **then**

SetSizeBtn.Enabled := true

**else**

SetSizeBtn.Enabled := false;

**end**;

**procedure** TMainF.EdgeEditKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **set of** char;

**begin**

Numerals := ['1'..'9', #8];

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

Key := #0;

**if** (Length(EdgeEdit.Text) = 1) **and** (Key <> #8) **then**

Key := #0;

**end**;

**procedure** TMainF.VertEditKeyPress(Sender: TObject; **var** Key: Char);

**var**

Numerals: **set of** char;

**begin**

Numerals := ['2'..'9', #8];

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

Key := #0;

**if** (Length(VertEdit.Text) = 1) **and** (Key <> #8) **then**

Key := #0;

**end**;

**end**.

**List.pas**

**unit** List;

**interface**

**uses**

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

System.Classes, Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs,

Vcl.Grids, Vcl.Menus;

**type**

TListF = **class**(TForm)

ListSG: TStringGrid;

PopupMenu: TPopupMenu;

MainMenu: TMainMenu;

SaveFile: TSaveDialog;

FileMenu: TMenuItem;

Save: TMenuItem;

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

**procedure** FormClose(Sender: TObject; **var** Action: TCloseAction);

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

**end**;

**var**

ListF: TListF;

**implementation**

{$R \*.dfm}

**procedure** TListF.FormClose(Sender: TObject; **var** Action: TCloseAction);

**var**

i, j: ShortInt;

**begin**

**for** j := 0 **to** ListSG.RowCount - 1 **do**

**for** i := 1 **to** ListSG.ColCount - 1 **do**

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

**end**;

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

**var**

i: ShortInt;

**begin**

**for** i := 1 **to** 9 **do**

ListSG.Cells[0, i - 1] := IntToStr(i);

ListSG.FixedCols := 1;

**end**;

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

**var**

OutputFile: TextFile;

MyFile: String;

ButtonSelected , i, j: byte;

**begin**

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

**begin**

MyFile := SaveFile.FileName;

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

**begin**

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

mtConfirmation, [mbYes,mbNo], 0);

AssignFile(OutputFile, MyFile);

**if** ButtonSelected = MrYes **then**

Rewrite(OutputFile)

**else**

**begin**

Append(outputFile);

WriteLn(OutputFile);

**end**;

**for** j := 0 **to** ListSG.RowCount - 1 **do**

**begin**

**for** i := 1 **to** ListSG.ColCount - 1 **do**

Write(OutputFile, ListSG.Cells[i, j]);

WriteLn(OutputFile);

**end**;

CloseFile(OutputFile);

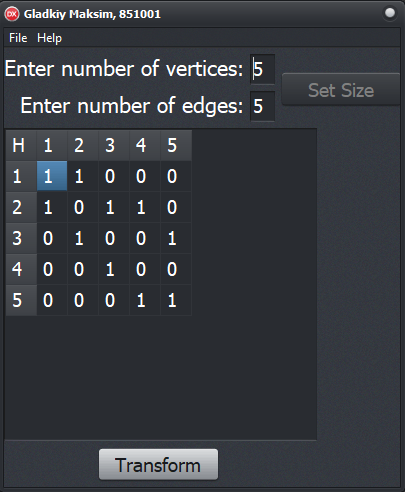
**end**;

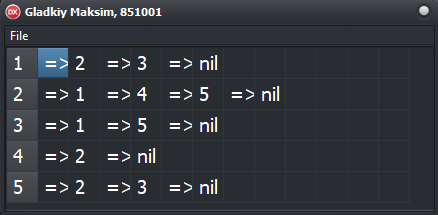
**end**;

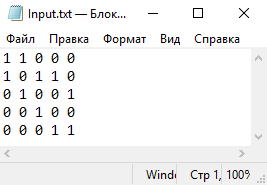
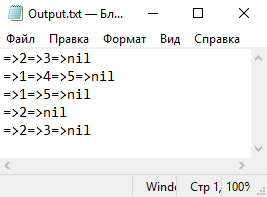
**end**;

**end**.

**Скриншоты:**

****

****

** **

**Блок-схема:**



