УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №7.2

по предмету

Основы алгоритмизации и программирования

Вариант 14

Выполнил:

Крутько А.А.

Проверила:

Данилова Г.В.

Группа 251004

Минск 2023

Задание:



Код программы на **Delphi**:

unit MainUnit;

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.NumberBox,

Vcl.Samples.Spin, Vcl.Buttons, ClipBRD, Vcl.ExtCtrls, Vcl.Menus;

type

TArrOI = Array of Array of Integer;

TIPEGrid = Class(TStringGrid);

TMainForm = class(TForm)

StringGrid1: TStringGrid;

SpinEdit1: TSpinEdit;

Label1: TLabel;

SpinEdit2: TSpinEdit;

Label2: TLabel;

BitBtn1: TBitBtn;

Timer1: TTimer;

StringGrid2: TStringGrid;

Label3: TLabel;

Label4: TLabel;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

OpenFile: TOpenDialog;

SaveFile: TSaveDialog;

procedure SpinEdit1Change(Sender: TObject);

procedure SpinEdit2Change(Sender: TObject);

procedure StringGrid1KeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

procedure StringGrid1SetEditText(Sender: TObject; ACol, ARow: Integer;

const Value: string);

procedure StringGrid1KeyPress(Sender: TObject; var Key: Char);

procedure Timer1Timer(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure N2Click(Sender: TObject);

private

public

end;

var

MainForm: TMainForm;

implementation

{$R \*.dfm}

Function IsGraphCorrect: Boolean;

Begin

Var I := 1;

Var IsCorrect := True;

While IsCorrect and (I < MainForm.StringGrid1.ColCount) do

Begin

Var Sum := 0;

For Var J := 1 to MainForm.StringGrid1.RowCount - 1 do

If MainForm.StringGrid1.Cells[I, J] <> '' Then

Inc(Sum, StrToInt(MainForm.StringGrid1.Cells[I, J]));

IsCorrect := Sum = 2;

Inc(I);

End;

Result := IsCorrect;

End;

Function GetAdjMatrix(Arr: TArrOI): TArrOI;

Var

ArrAdj: TArrOI;

I, J, SecondIndex: Integer;

Begin

SetLength(ArrAdj, Length(Arr), Length(Arr));

For I := 0 to High(Arr[0]) do

Begin

J := 0;

While Arr[J, I] = 0 do

Inc(J);

SecondIndex := J + 1;

While (SecondIndex < Length(Arr)) and (Arr[SecondIndex,I] = 0) do

Inc(SecondIndex);

If SecondIndex = Length(Arr) Then

Inc(ArrAdj[J,J])

Else

Begin

Inc(ArrAdj[J, SecondIndex]);

Inc(ArrAdj[SecondIndex, J]);

End;

End;

GetAdjMatrix := ArrAdj;

End;

procedure TMainForm.BitBtn1Click(Sender: TObject);

Var

Arr1,Arr: TArrOI;

I, J: Integer;

begin

For I := 1 to StringGrid2.RowCount - 1 do

StringGrid2.Rows[I].Clear;

StringGrid2.ColCount := SpinEdit2.Value + 1;

StringGrid2.RowCount := SpinEdit2.Value + 1;

SetLength(Arr, StringGrid1.RowCount - 1, StringGrid1.ColCount - 1);

I := 1;

While I < StringGrid1.RowCount do

Begin

J := 1;

While J < StringGrid1.ColCount do

Begin

If StringGrid1.Cells[J, I] <> '' Then

Arr[I - 1, J - 1] := StrToInt(StringGrid1.Cells[J, I])

Else

Arr[I - 1, J - 1] := 0;

Inc(J);

End;

Inc(I);

End;

Arr1 := GetAdjMatrix(Arr);

For I := 0 to High(Arr1) do

For J := 0 to High(Arr1[0]) do

StringGrid2.Cells[J + 1, I + 1] := IntToStr(Arr1[I, J]);

N5.Enabled := True;

StringGrid2.Visible := True;

end;

procedure TMainForm.N2Click(Sender: TObject);

begin

ShowMessage('Данная программа преобразует матрицу инцидентности графа в матрицу   
 смежности'#13#10 +

'1. Сумма в каждом столбце должна быть равна 2'#13#10 +

'2. 2 в ячейке образует петлю у данной вершины');

end;

procedure TMainForm.N3Click(Sender: TObject);

begin

ShowMessage('Крутько Андрей 251004');

end;

procedure Split(Delimiter: Char; Str: string; ListOfStrings: TStringList); StdCall;

begin

ListOfStrings.Clear;

ListOfStrings.Delimiter := Delimiter;

ListOfStrings.StrictDelimiter := True;

ListOfStrings.DelimitedText := Str;

end;

procedure TMainForm.N4Click(Sender: TObject);

Var

FileInput: TextFile;

Temp: String;

ListTemp: TStringList;

Arr: TArrOI;

I, J, Count, Sum, SizeRow, SizeCol: Integer;

Begin

If OpenFile.Execute Then

Begin

Try

Try

AssignFile(FileInput, ChangeFileExt(OpenFile.FileName, '.txt'));

Reset(FileInput);

Readln(FileInput, Temp);

SizeRow := StrToInt(Temp);

If (SizeRow < 1) or (SizeRow > 10) Then

raise Exception.Create('Проверьте правильность размера   
 массива');

Readln(FileInput, Temp);

SizeCol := StrToInt(Temp);

If (SizeCol < 1) or (SizeCol > 10) Then

raise Exception.Create('Проверьте правильность размера   
 массива');

SetLength(Arr, SizeRow, SizeCol);

ListTemp := TStringList.Create;

J := 0;

While Not Eof(FileInput) do

Begin

Readln(FileInput, Temp);

Split(' ', Temp, ListTemp);

If ListTemp.Count = SizeCol Then

Begin

For I := 0 to ListTemp.Count - 1 do

Arr[J, I] := StrToInt(ListTemp[I]);

End

Else

raise Exception.Create('Несовпадение элементов с   
 размерностью списка');

Inc(J);

End;

For I := 0 to High(Arr[0]) do

Begin

Sum := 0;

For J := 0 to High(Arr) do

Inc(Sum, Arr[J, I]);

If Sum <> 2 Then

raise Exception.Create('Сумма по столбцам не везде равна   
 2');

End;

StringGrid1.ColCount := SizeCol + 1;

StringGrid1.RowCount := SizeRow + 1;

SpinEdit1.Value := SizeCol;

SpinEdit2.Value := SizeRow;

For I := 0 to High(Arr) do

For J := 0 to High(Arr[0]) do

StringGrid1.Cells[J + 1, I + 1] := IntToStr(Arr[I, J]);

BitBtn1.Enabled := True;

Except

on E : Exception do

Begin

ShowMessage('Поднята ошибка, с сообщением : ' +   
 E.Message);

Arr := Nil;

End;

End;

Finally

CloseFile(FileInput);

End;

End;

end;

procedure TMainForm.N5Click(Sender: TObject);

Var

FileOutput: TextFile;

begin

If SaveFile.Execute Then

Begin

Try

Try

AssignFile(FileOutput, ChangeFileExt(SaveFile.FileName, '.txt'));

ReWrite(FileOutput);

For Var I := 0 to StringGrid2.RowCount - 1 do

Begin

For Var J := 0 to StringGrid2.ColCount - 1 do

Write(FileOutput, StringGrid2.Cells[J, I]:4);

Writeln(FileOutput);

End;

Except

On E: Exception do

ShowMessage('Проверьте файл, ошибка ' + E.Message);

End;

Finally

CloseFile(FileOutput);

End;

End;

end;

procedure TMainForm.SpinEdit1Change(Sender: TObject);

begin

If SpinEdit1.Value + 1 < StringGrid1.ColCount Then

StringGrid1.Cols[StringGrid1.ColCount - 1].Clear;

StringGrid1.ColCount := SpinEdit1.Value + 1;

BitBtn1.Enabled := CheckGrid;

StringGrid2.Visible := False;

end;

procedure TMainForm.SpinEdit2Change(Sender: TObject);

begin

If SpinEdit2.Value + 1 < StringGrid1.RowCount Then

StringGrid1.Rows[StringGrid1.RowCount - 1].Clear;

StringGrid1.RowCount := SpinEdit2.Value + 1;

BitBtn1.Enabled := CheckGrid;

StringGrid2.Visible := False;

end;

procedure TMainForm.StringGrid1SetEditText(Sender: TObject; ACol, ARow: Integer;

const Value: string);

begin

BitBtn1.Enabled := IsGraphCorrect;

StringGrid2.Visible := False;

end;

procedure TMainForm.Timer1Timer(Sender: TObject);

begin

For Var I := 1 to MainForm.StringGrid1.RowCount - 1 do

MainForm.StringGrid1.Cells[0,I] := IntToStr(I);

For Var I := 1 to MainForm.StringGrid1.ColCount - 1 do

MainForm.StringGrid1.Cells[I,0] := Chr(I + 96);

For Var I := 1 to MainForm.StringGrid2.RowCount - 1 do

Begin

MainForm.StringGrid2.Cells[0,I] := IntToStr(I);

MainForm.StringGrid2.Cells[I,0] := IntToStr(I);

End;

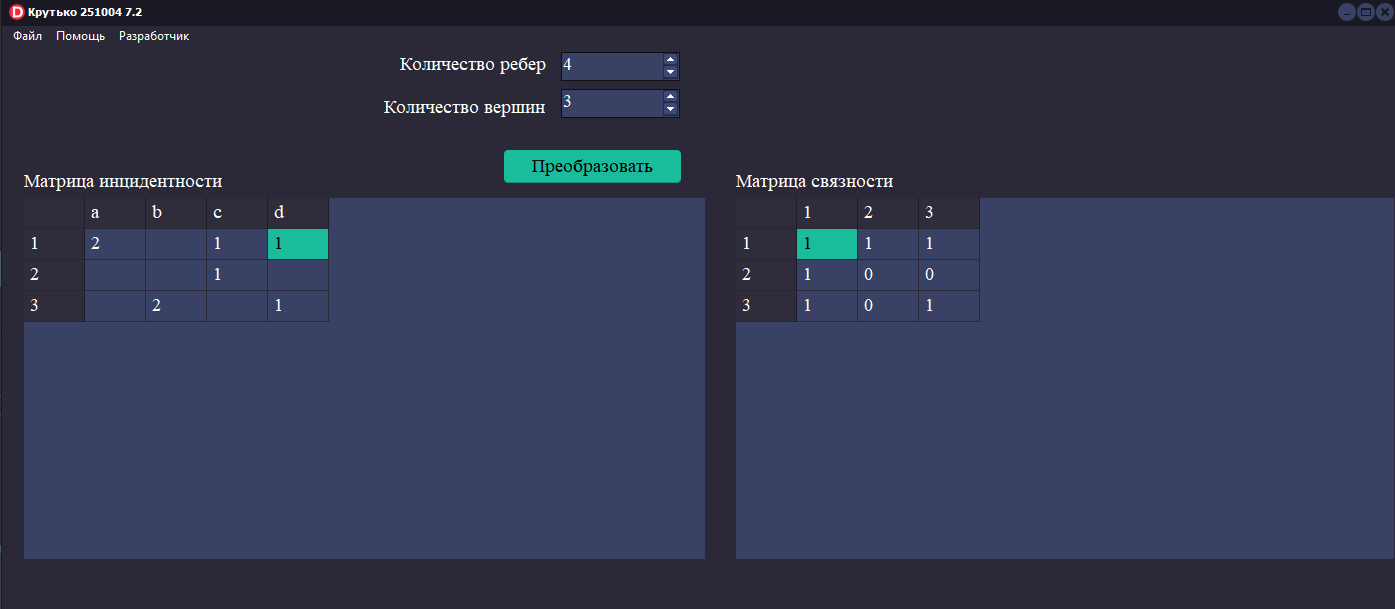
end;

end.

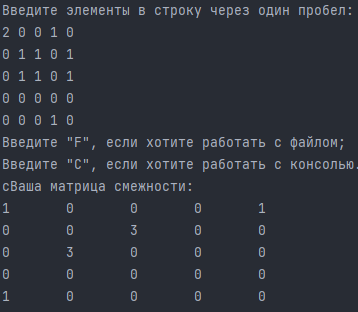
Код программы на **C#**:

Console.WriteLine("Данная программа преобразует матрицу инцидентности графа в матрицу смежности");  
int[,] arrAdj = GetKey().Key == ConsoleKey.C  
 ? GetAdjMatrix(InputInfo(GetNumber(1, 10), GetNumber(1, 10)))  
 : GetAdjMatrixFile();  
if (GetKey().Key == ConsoleKey.C)  
{  
 Console.WriteLine("Ваша матрица смежности: ");  
 Print(Console.Write, arrAdj);  
}  
else  
 SaveSquareToFile(arrAdj);  
  
int[,] GetAdjMatrix(int[,] arr)  
{  
 int[,] arrAdj = new int[arr.GetLength(0), arr.GetLength(0)];  
 for (int i = 0; i < arr.GetLength(1); i++)  
 {  
 int j = 0;  
 while (arr[j, i] == 0)  
 j++;  
 int secondIndex = j + 1;  
 while (secondIndex < arr.GetLength(0) && arr[secondIndex, i] == 0)  
 secondIndex++;  
 if (secondIndex == arr.GetLength(0))  
 arrAdj[j, j]++;  
 else  
 {  
 arrAdj[j, secondIndex]++;  
 arrAdj[secondIndex, j]++;  
 }  
 }  
 return arrAdj;  
}  
  
void SaveSquareToFile(int[,] arrAdj)  
{  
 bool isIncorrect;  
 do {  
 isIncorrect = true;  
 try  
 {  
 using (var sw = new StreamWriter(TakeFilePath(), false))  
 {  
 sw.WriteLine("Ваша матрица смежности: ");  
 Print(sw.Write, arrAdj);  
 }  
 }  
 catch (Exception e)  
 {  
 isIncorrect = false;  
 Console.WriteLine(e.Message);  
 }  
 } while (!isIncorrect);  
}  
   
int[,] GetAdjMatrixFile()  
{  
 int[,] arrNew = {};  
 bool isIncorrect;  
 do  
 {  
 isIncorrect = true;  
 try  
 {  
 using (var st = new StreamReader(TakeFilePath()))  
 {  
 List<List<int>> arr = new List<List<int>>();  
 int sizeRow = int.Parse(st.ReadLine());  
 if (sizeRow > 10 || sizeRow < 1)  
 throw new Exception("Длина массива не соответствует валидной");  
 int sizeCol = int.Parse(st.ReadLine());  
 if (sizeCol > 10 || sizeCol < 1)  
 throw new Exception("Длина массива не соответствует валидной");  
 string strTemp = "";  
 while (!string.IsNullOrEmpty(strTemp = st.ReadLine()))  
 {  
 var strArr = strTemp.Split(' ');  
 if (strArr.Length != sizeCol)  
 {  
 isIncorrect = false;  
 Console.WriteLine("Проверьте ввод");  
 }  
 List<int> arrTemp = new List<int>();  
 for (int i = 0; i < strArr.Length && isIncorrect; i++)  
 {  
 isIncorrect = int.TryParse(strArr[i], out var temp);  
 if (!isIncorrect)  
 Console.WriteLine("Проверьте ввод в файле");  
 else  
 arrTemp.Add(temp);  
 }  
 arr.Add(arrTemp);  
 }  
 if (arr[^1].Count != sizeCol)  
 {  
 isIncorrect = false;  
 Console.WriteLine("Проверьте ввод");   
 }  
 arrNew = new int[sizeRow, sizeCol];  
 int numRows = arr.Count;  
 int numCols = arr[numRows - 1].Count;  
 for (int i = 0; i < numRows; i++)  
 for (int j = 0; j < numCols; j++)  
 arrNew[i, j] = arr[i][j];  
 if (isIncorrect)  
 {  
 for (int i = 0; i < arrNew.GetLength(1) && isIncorrect; i++)  
 {  
 int sum = 0;  
 for (int j = 0; j < arrNew.GetLength(0) && isIncorrect; j++)  
 {  
 sum += arrNew[j, i];  
 if (arrNew[j, i] < 0 || arrNew[j, i] > 2)  
 {  
 isIncorrect = false;  
 Console.WriteLine("Сумма ваших элементов по столбцам   
 должна быть равна 2");  
 }  
 }  
  
 if (sum != 2)  
 {  
 isIncorrect = false;  
 Console.WriteLine("Сумма ваших элементов по столбцам   
 должна быть равна 2");  
 }  
 }  
 }  
 }  
   
 }  
 catch (Exception e)  
 {  
 Console.WriteLine(e.Message);  
 }  
 } while (!isIncorrect);  
 return GetAdjMatrix(arrNew);  
}  
  
string? TakeFilePath()  
{  
 string? path;  
 bool isIncorrect;  
 do {  
 isIncorrect = false;  
 Console.WriteLine("Введите путь к файлу");  
 path = Console.ReadLine();  
 if (!File.Exists(path) || !path.EndsWith(".txt"))  
 {  
 isIncorrect = true;  
 Console.WriteLine("Проверьте параметры файла");  
 }  
 } while (isIncorrect);  
 return path;  
}  
  
int GetNumber(in int min, in int max)  
{  
 int numb;  
 do {  
 } while (!int.TryParse(Console.ReadLine(), out numb) || numb < min || numb > max);  
 return numb;  
}  
  
ConsoleKeyInfo GetKey()  
{  
 ConsoleKeyInfo key;  
 Console.WriteLine("Введите \"F\", если хотите работать с файлом;\nВведите \"C\", если хотите работать с консолью.");  
 do {  
 key = Console.ReadKey();  
 } while (key.Key != ConsoleKey.C && key.Key != ConsoleKey.F);  
 return key;  
}  
  
int[,] InputInfo(int sizeRow, int sizeCol)  
{  
 int[,] arr = new int[sizeRow, sizeCol];  
 Console.WriteLine("Введите элементы в строку через один пробел:");  
 bool isInCorrect;  
 do {  
 isInCorrect = true;  
 for (int i = 0; i < arr.GetLength(0) && isInCorrect; i++)  
 {  
 var strArr = Console.ReadLine().Split(' ');  
 if (strArr.Length != arr.GetLength(1))  
 {  
 isInCorrect = false;  
 Console.WriteLine("Проверьте ввод");  
 }  
  
 for (int j = 0; j < strArr.Length && isInCorrect; j++)  
 {  
 if (!int.TryParse(strArr[j], out arr[i, j]))  
 Console.WriteLine("Проверьте ввод");  
 }  
 }  
 if (isInCorrect)  
 {  
 for (int i = 0; i < arr.GetLength(1) && isInCorrect; i++)  
 {  
 int sum = 0;  
 for (int j = 0; j < arr.GetLength(0) && isInCorrect; j++)  
 {  
 sum += arr[j, i];  
 if (arr[j, i] < 0 || arr[j, i] > 2)  
 {  
 isInCorrect = false;  
 Console.WriteLine("Сумма ваших элементов по столбцам должна   
 быть равна 2");   
 }  
 }  
 if (sum != 2)  
 {  
 isInCorrect = false;  
 Console.WriteLine("Сумма ваших элементов по столбцам должна быть   
 равна 2");  
 }  
 }  
 }  
 } while (!isInCorrect);  
 return arr;  
}  
  
void Print(Action<object> action, int[,] arr)  
{  
 for (int i = 0; i < arr.GetLength(0); i++) {  
 for (int j = 0; j < arr.GetLength(1); j++)  
 action(arr[i,j] + "\t");  
 action("\n");  
 }  
}

Результат на **Delphi**:



Результат на **C#**:



**Блок-cхема**:

