Министерство Науки, Образования, Молодежи и Спорта  
Одесская Национальная Академия Пищевых Технологий  
Институт Холода, Криотехнологий и Экоэнергетики  
Факультет Информационных Технологий и Кибербезопасности

**Индивидуальное задание**

По дисциплине:

ЧМИ

Выполнил:

Парфенюк А.Э.

Проверил:

Антонова А. Р.

1 Разработать на языке программирование “Object Рascal” модуль, который содержит:

- описание класса данных для хранение двумерного массива (матрицы);

-методы для чтения из файла и сохранения в файл описанного класаса;

- описание дополнительных классов для хранения результатов.

2 Разработать на языке программирования “Object Рascal” модуль обработки двумерных массивов, который содержит процедуры или функции, выполняющие расчет в соответствии с заданиями, приведенными на обороте.

- имена параметров процедур (функций) должны соответствовать заданию;

- процедуры и функции должны генерировать исключения в случае невозможности определить результат обработки массива;

3 Разработать в среде «» оконное Windows приложение, использующее модуль см, п1, и отвечающее следующим требованием:

3.1 Приложение должно обеспечивать :

- Создание нового набора данных;

-ввод и редактирование размеров и значений двухмерного массива описанного структурой см. п1. Ввод размеров и значений элементов массива производить в отдельном окне;

-Чтение из файла, сохранение в файл информации о массиве;

-Выполнение обработки массива по заданиям п.1;

-Сохранение в текстовый файл результатов обработки и если пользователю необходимо, исходных данных в табличной форме.

3.2 Дополнительные требования:

-При работе с файлом использовать только стандартные диалоги открытия и сохранения;

-Главное окно приложения должно содержать главное меню со всеми командами, реализованными в приложении;

-Приложение должно содержать окно выводящее информацию о разработчике.

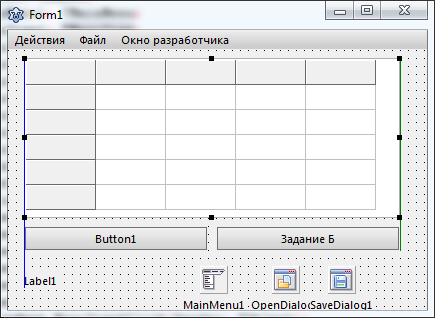
Вариант 11

Задание:для двухмерного массива N(4\*6),что составлен с действительных чисел,найти:

А)Суммы елементов,размещенных в каждом с 4-ох квадратов,если условно поделить матрицу на 4 равные части;

Б)Составить одномерный массив В из елементов, которые являются последними парными

елементами в каждом рядке массива А;

**unit Unit1;**

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, Grids,

StdCtrls, Menus,matrica,okno\_razr,proced,rezult;

type

{ TForm1 }

TForm1 = class(TForm)

Button1: TButton;

Button2: TButton;

Label1: TLabel;

MainMenu1: TMainMenu;

MenuItem1: TMenuItem;

MenuItem2: TMenuItem;

MenuItem3: TMenuItem;

MenuItem4: TMenuItem;

MenuItem5: TMenuItem;

MenuItem6: TMenuItem;

MenuItem7: TMenuItem;

MenuItem8: TMenuItem;

OpenDialog1: TOpenDialog;

SaveDialog1: TSaveDialog;

StringGrid1: TStringGrid;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure MenuItem3Click(Sender: TObject);

procedure MenuItem4Click(Sender: TObject);

procedure MenuItem5Click(Sender: TObject);

procedure MenuItem6Click(Sender: TObject);

procedure MenuItem7Click(Sender: TObject);

procedure MenuItem8Click(Sender: TObject);

procedure ShowMt(MTR:Matrix;c: word; r: word);

function ReadMt:Matrix;

private

{ private declarations }

public

{ public declarations }

end;

var

Form1: TForm1;

Matr1: TMatrix;

Res1: TResult;

implementation

{$R \*.lfm}

{ TForm1 }

**procedure TForm1.MenuItem4Click(Sender: TObject);**

var

s:string;

i,j,r,c,code:word;

begin

randomize;

StringGrid1.Clean;

s:=InputBox('Creating of matrix','Введите число колонок:','4');

val(s,c,code);

if (code<>0) or (c<=0)

then showmessage('Ошибочное значение')

else

begin

s:=InputBox('Creating of matrix','Введите число рядов:','6');

val(s,r,code);

if (code<>0) or (r<=0)

then showmessage('Ошибочное значение')

else

begin

with StringGrid1 do

begin

ColCount:=c+1;

RowCount:=r+1;

for i:=1 to c do

for j:=1 to r do

Cells[i,j]:=FloatToStr(Random(300)-150);

Cells[0,0]:='Matrix';

for i:=1 to ColCount-1 do

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

for i:=1 to RowCount-1 do

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

end;

Button1.Enabled:=true;

Button2.Enabled:=true;

end;

end;

Button1.Enabled:=True;

Button2.Enabled:=True;

MenuItem6.Enabled:=True;

end;

**procedure TForm1.Button1Click(Sender: TObject);**

var

i:word;

Mas:Massiv;

str:string;

begin

str:='';

if (Length(ReadMt) mod 2<>0) or (Length(ReadMt[0]) mod 2<>0)

then

begin

Showmessage('Размеры матрицы должны быть чётными числами!');

exit;

end

else

Mas:=SumElem(ReadMt);

for i:=0 to Length(Mas)-1 do

str:=str+FloattoStr(i+1)+' : '+Floattostr(Mas[i])+' ';

Label1.Caption:='Результаты А : '+str;

//Showmessage.Title:='Суммы квадратов';

Showmessage('Результаты А : '+str);

end;

**procedure TForm1.Button2Click(Sender: TObject);**

var

icol,irow:integer;

i:word;

m:Massiv2;

str:string;

begin

str:='';

m:=LastParn(ReadMt);

Label1.Caption:='Успешно';

Form3:=TForm3.Create(Form1);

Form3.Visible:=true;

Form3.RezB(Length(m),m);

end;

**procedure TForm1.FormCreate(Sender: TObject);**

begin

StringGrid1.ColCount:=5;

StringGrid1.RowCount:=6;

Button1.Enabled:=False;

Button2.Enabled:=False;

MenuItem6.Enabled:=False;

end;

**procedure TForm1.MenuItem3Click(Sender: TObject);**

begin

Form2.Show;

end;

**procedure TForm1.MenuItem5Click(Sender: TObject);**

begin

try

if OpenDialog1.Execute then

begin

StringGrid1.Clean;

Matr1:=TMatrix.Create;

Matr1.Open(OpenDialog1.FileName);

ShowMt(Matr1.Mt,Matr1.FCol,Matr1.FRow);

Matr1.Free;

Button1.Enabled:=true;

Button2.Enabled:=true;

MenuItem6.Enabled:=True;

end;

except

on EAccessViolation do

begin

showmessage('Ошибка чтения файла: файл поврежден');

exit;

end;

on EConvertError do

begin

showmessage('Ошибка чтения файла: файл поврежден');

exit;

end;

end;

end;

**procedure TForm1.MenuItem6Click(Sender: TObject);**

begin

if SaveDialog1.Execute then

begin

try

Res1:=TResult.Create;

Res1.Mas:=SumElem(ReadMt);

Res1.Mas2:=LastParn(ReadMt);

Res1.Save(SaveDialog1.FileName);

Res1.Free;

Showmessage('Сохранено');

except

on EConvertError do

begin

Showmessage('Массив пустой!!');

exit;

end;

end;

end;

end;

**procedure TForm1.MenuItem7Click(Sender: TObject);**

begin

if SaveDialog1.Execute then

begin

try

Matr1:=TMatrix.Create;

Matr1.Mt:=ReadMt;

Matr1.save(SaveDialog1.FileName);

showmessage('Матрица сохранена!');

Matr1.Free;

except

on EConvertError do

begin

showmessage('ConvertError(Неверно введен массив)');

exit;

end;

end;

end;

end;

**procedure TForm1.MenuItem8Click(Sender: TObject);**

begin

Close;

end;

**procedure TForm1.ShowMt(MTR: Matrix; c: word; r: word);**

var

i,j:word;

begin

StringGrid1.ColCount:=c+1;

StringGrid1.RowCount:=r+1;

StringGrid1.Cells[0,0]:='Matrix';

for i:=1 to StringGrid1.ColCount-1 do

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

for i:=1 to StringGrid1.RowCount-1 do

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

for i:=0 to StringGrid1.ColCount-2 do

for j:=0 to StringGrid1.RowCount-2 do

begin

StringGrid1.Cells[i+1,j+1]:=FloatToStr(MTR[j,i]);

end;

end;

**function TForm1.ReadMt:Matrix;**

var

c,r:word;

begin

for c:=1 to StringGrid1.ColCount-1 do

begin

SetLength(result,c);

for r:=1 to StringGrid1.RowCount-1 do

begin

SetLength(result[c-1],r);

result[c-1,r-1]:=StrToInt(StringGrid1.Cells[c,r]); end; end;

end;

end.

**unit okno\_razr;**

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls;

type

{ TForm2 }

TForm2 = class(TForm)

Button2: TButton;

Memo1: TMemo;

procedure Button2Click(Sender: TObject);

private

{ private declarations }

public

{ public declarations }

end;

var

Form2: TForm2;

implementation

{$R \*.lfm}

{ TForm2 }

**procedure TForm2.Button2Click(Sender: TObject);**

begin

Close;

end;

end.

**unit matrica;**

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Dialogs;

type

Matrix=array of array of real;

Massiv=array of real;

Massiv2=array of integer;

{ TMatrix}

**TMatrix=class**

public

Mt:Matrix;

FCol:word;

FRow:word;

procedure Open(FileN:string);

procedure Save(FileN:string);

end;

{ TResult }

**TResult=class**

public

Mt:Matrix;

Mas:massiv;

Mas2:massiv2;

procedure Save(FileN:string);

end;

implementation

{ TMatrix }

**procedure TMatrix.Open(FileN:string);**

var

F:Text;

i,col,row:word;

str,sym:string;

begin

AssignFile(F,FileN);

Reset(F);

row:=-1;

while not Eof(F) do

begin

Inc(Row);

SetLength(Mt,row+1);

col:=-1;

readln(F,str);

for i:=1 to length(str) do

begin

if str[i]<>' '

then

begin

sym:=sym+str[i];

end

else

begin

Inc(col);

SetLength(Mt[row],col+1);

Mt[row,col]:=StrToInt(sym);

sym:='';

end;

end;

end;

FRow:=row+1;

FCol:=col+1;

CloseFile(F);

end;

**procedure TMatrix.Save(FileN:string);**

var

F:TextFile;

n,m,i,j:integer;

begin

n:=Length(Mt);

m:=Length(Mt[0]);

AssignFile(F,FileN);

Rewrite(F);

for i:=0 to m-1 do

begin

for j:=0 to n-1 do

Write(F,Floattostr(Mt[j,i])+' ');

writeln(F);

end;

CloseFile(F);

end;

{ TResult }

**procedure TResult.Save(FileN:string);**

var

F:TextFile;

Col,Row,c:integer;

str:string;

begin

AssignFile(F,FileN);

Rewrite(F);

writeln(F,'Задание А');

Col:=Length(Mas);

for c:=0 to Col-1 do

str:=str+InttoStr(c+1)+': '+Floattostr(Mas[c])+' ; ';

writeln(F,'Результаты:'+str);

writeln(F,'Задание Б');

writeln(F,'Последние непарные елементы рядов:');

Row:=Length(Mas2);

for c:=0 to Row-1 do

writeln(F,InttoStr(c+1)+': '+InttoStr(Mas2[c]));

CloseFile(F);

end;

end.

**unit proced;**

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Matrica, Dialogs;

function SumElem(Mt:Matrix):massiv;

function LastParn(Mt:Matrix):massiv2;

implementation

**function SumElem(Mt:Matrix):massiv;**

var ai:array[0..3]of real;

Rows,Cols,i,j:integer;

sum:real;

begin

Rows:=Length(Mt)-1;

Cols:=Length(Mt[0])-1;

sum:=0;

for j:=0 to Cols div 2 do

for i:=0 to Rows div 2 do

sum:=Sum+Mt[i,j];

ai[0]:=sum;

sum:=0;

for j:=0 to (Cols div 2) do

for i:=(Rows div 2)+1 to Rows do

sum:=Sum+Mt[i,j];

ai[1]:=sum;

sum:=0;

for j:=(Cols div 2)+1 to Cols do

for i:=0 to (Rows div 2) do

sum:=Sum+Mt[i,j];

ai[2]:=sum;

sum:=0;

for j:=(Cols div 2)+1 to Cols do

for i:=(Rows div 2)+1 to Rows do

sum:=Sum+Mt[i,j];

ai[3]:=sum;

result:=ai;

end;

**function LastParn(Mt:Matrix):massiv2;**

var

a:Massiv2;

Rows,Cols:integer;

i,j:word;

Cl:word;

flag:boolean;

begin

Rows:=Length(Mt);

Cols:=Length(Mt[0]);

SetLength(a,Length(Mt[0]));

Cl:=-1;

for j:=0 to Cols do

begin

flag:=false;

i:=Rows;

While (i>0)and(not(flag)) do

begin

if not Odd(trunc(Mt[i,j])) then

begin

flag:=true;

inc(cl);

a[Cl]:=(trunc(Mt[i,j]));

end;

i:=i-1;

end;

if not(flag) then

begin

inc(cl);

a[Cl]:=null;

end;

end;

result:=a;

end;

end.

**unit rezult;**

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, Grids,

StdCtrls, proced, matrica;

type

{ TForm3 }

TForm3 = class(TForm)

Button1: TButton;

StringGrid1: TStringGrid;

procedure Button1Click(Sender: TObject);

procedure RezB(r:word;m:Massiv2);

private

{ private declarations }

public

{ public declarations }

end;

var

Form3: TForm3;

implementation

{$R \*.lfm}

{ TForm3 }

**procedure TForm3.Button1Click(Sender: TObject);**

begin

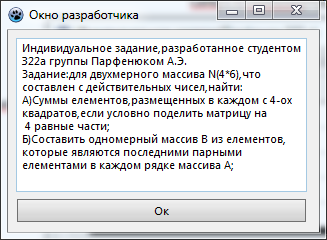
Close;

end;

**procedure TForm3.RezB(r:word;m:Massiv2);**

var

i:word;

begin

with StringGrid1 do

begin

RowCount:=r+1;

Cells[0,0]:='Последние непарные';

Cells[1,0]:='Col 1';

for i:=1 to RowCount-1 do

begin

Cells[0,i]:='Ряд '+FloatToStr(i);

Cells[1,i]:=InttoStr(m[i]);

end;

end;

end;

end.

