**Практическая работа №6**  
**«Двумерные массивы с DLL-библиотеками »**

В данной практической работе нам необходимо использовать двумерные массивы, одномерные динамические массивы, функции и процедуры, которые находятся в DLL-библиотеке:

1. Сформировать двумерный массив используя процедуру ввода и вывода, в DDL-библиотеке.
2. Разработать функцию в DDL-библиотеке, которая будет возвращать сумму отрицательных элементов двумерного массива.
3. Разработать процедуру в DDL-библиотеке, которая формирует одномерный массив из тех элементов исходного двумерного массива, которые не являются четными.

Ниже представлен код DLL-библиотеки с процедурой libr\_summ\_otr:

library libr\_summ\_otr;

{$mode objfpc}{$H+}

uses

Classes

{ you can add units after this };

type dvmas = array [1..15,1..15] of integer;

function summ\_otr(a:dvmas; n,m:integer):integer;

var i,j:integer;

begin

summ\_otr:= 0;

for i:=1 to n do

begin

for j:=1 to m do

begin

if(a[i,j] < 0) then

summ\_otr:= summ\_otr + a[i,j];

end;

end;

end;

exports summ\_otr;

begin

end.

Ниже представлен код DLL-библиотеки libr\_form\_dvmas:

library libr\_form\_dvmas;

{$mode objfpc}{$H+}

uses

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

Grids, Interfaces;

type dvmas = array [1..15,1..15] of integer;

procedure form\_dvmas(var a:dvmas; var n,m:integer; var s1 :TStringGrid);

var i,j :integer;

begin

randomize;

for i:=1 to n do

begin

for j:=1 to m do

begin

a[i,j]:= -10 + random(55);

end;

end;

with s1 do

begin

colcount:= m + 1;

rowcount:= n + 1;

for i:=1 to rowcount -1 do

cells[0,i]:= inttostr(i);

for j:=1 to colcount -1 do

cells[j,0]:= inttostr(j);

for i:=1 to n do

for j:=1 to m do

cells[i,j]:= inttostr(a[i,j]);

end;

end;

exports form\_dvmas;

begin

end.

Ниже представлен код DLL-библиотеки libr\_rezmas:

library libr\_rezmas;

{$mode objfpc}{$H+}

uses

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

Grids, Interfaces;

type dvmas = array [1..15,1..15] of integer;

type mas = array of integer;

procedure rezmas(var a:dvmas;var rez:mas; var n,m:integer; var s2 :TStringGrid);

var i,j,k:integer;

begin

k:=0;

setlength(rez,1);

for i:=1 to n do

begin

for j:=1 to m do

begin

if(a[i,j] mod 2 <> 0) then

begin

rez[k]:= a[i,j];

k:= k+1;

setlength(rez,k+1);

end;

end;

end;

with s2 do

begin

for i:=low(rez) to high(rez) do

begin

cells[0,i]:= inttostr(i);

cells[1,i]:= inttostr(rez[i]);

rowcount:= rowcount + 1;

end;

rowcount:= rowcount - 1;

end;

end;

exports rezmas;

begin

end.

Ниже представлен код основной программы:

implementation

type mas = array of integer;

type dvmas = array [1..15,1..15] of integer;

function summ\_otr(a:dvmas; n,m:integer):integer;

external 'libr\_summ\_otr.dll' name 'summ\_otr';

procedure form\_dvmas(var a:dvmas; var n,m:integer; var s1 :TStringGrid );

external 'libr\_form\_dvmas.dll' name 'form\_dvmas';

procedure rezmas(var a:dvmas;var rez:mas; var n,m:integer; var s2 :TStringGrid);

external 'libr\_rezmas.dll' name 'rezmas';

{$R \*.lfm}

{ TForm1 }

procedure TForm1.Button1Click(Sender: TObject);

var rez:mas;

a:dvmas;

n,m,summ:integer;

begin

n:=strtoint(edit1.text);

m:=strtoint(edit2.text);

form\_dvmas(a,n,m,StringGrid1);

summ:= summ\_otr(a,n,m);

showmessage('Сумма отрицательных чисел ' + inttostr(summ));

rezmas(a,rez,n,m,StringGrid2);

end;

На Рисунках 1 и 2 представлены результаты работы программы:

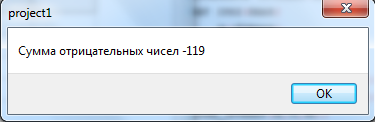


Рисунок 1 - Промежуточный результат работы программы

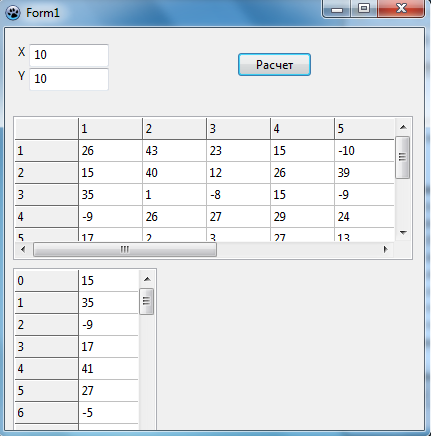


Рисунок 2 - Окончательный результат работы программы