**Практическая работа №7**  
**«Работа с текстовыми и двоичными файлами»**

В данной практической работе нам необходимо использовать двумерные массивы, одномерные динамические массивы, функции и процедуры, которые находятся в DLL-библиотеке, также нам необходимо провести работу с текстовыми и бинарными файлами для решения следующей задачи:

1. Сформировать двумерный массив используя процедуру ввода и вывода, в DDL-библиотеке, так же записать его в бинарный и текстовый файл.
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, а также записать его в бинарный и текстовый файл.

С помощью двух кнопок нужно разделить текстовый и бинарный вариант работы.

Ниже представлены коды DDL-библиотек для работы с текстовыми файлами:

**libr\_rezmas\_txt.dll**

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 rez:mas; var n,m:integer; var s2 :TStringGrid; var fname:string);

var a:dvmas;

var i,j,k:integer;

var tf :TextFile;

buff:string;

begin

AssignFile(tf,fname);

Reset(tf);

k:=0;

setlength(rez,1);

for i:=1 to n do

for j:=1 to m do

begin

Readln(tf,buff);

a[i,j]:= strtoint(buff);

end;

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;

writeln(tf,000000000000);

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

begin

writeln(tf,rez[i]);

end;

Closefile(tf);

end;

exports rezmas;

begin

end.

**libr\_form\_dvmas\_txt.dll**

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 fname:string);

var i,j :integer;

var tf :TextFile;

begin

AssignFile(tf,fname);

Rewrite(tf);

randomize;

for i:=1 to n do

begin

for j:=1 to m do

begin

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

end;

end;

for i:=1 to n do

begin

for j:=1 to m do

begin

writeln(tf,inttostr(a[i,j]))

end;

end;

CloseFile(tf);

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.

**libr\_summ\_otr\_txt.dll**

library libr\_summ\_otr;

{$mode objfpc}{$H+}

uses

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

Grids, Interfaces;

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

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

var i,j,zps:integer;

var tf :TextFile;

begin

AssignFile(tf,fname);

Append(tf);

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;

writeln(tf,'Сумма отрицательных чисел = ' + inttostr(summ\_otr));

closefile(tf);

end;

exports summ\_otr;

begin

end.

Ниже представлены коды DDL-библиотек для работы с бинарными файлами:

**libr\_rezmas.dll**

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\_bin(var rez:mas; var n,m:integer; var s2 :TStringGrid; var fname:string);

var a:dvmas;

var i,j,k:integer;

var tf :File;

begin

AssignFile(tf,fname);

Reset(tf,1);

k:=0;

setlength(rez,1);

for i:=1 to n do

begin

for j:=1 to m do

begin

Blockread(tf,a[i,j],sizeof(integer));

end;

end;

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;

Closefile(tf);

end;

exports rezmas\_bin;

begin

end.

**libr\_form\_dvmas.dll**

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\_bin(var a:dvmas; var n,m:integer; var s1 :TStringGrid; var fname:string);

var i,j :integer;

var tf :File;

begin

AssignFile(tf,fname);

Rewrite(tf,1);

randomize;

for i:=1 to n do

begin

for j:=1 to m do

begin

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

end;

end;

for i:=1 to n do

begin

for j:=1 to m do

begin

Blockwrite(tf,a[i,j],sizeof(integer))

end;

end;

CloseFile(tf);

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\_bin;

begin

end.

**libr\_summ\_otr.dll**

library libr\_summ\_otr;

{$mode objfpc}{$H+}

uses

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

Grids, Interfaces;

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

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

var i,j:integer;

var tf :File;

begin

AssignFile(tf,fname);

Reset(tf,1);

summ\_otr\_bin:= 0;

for i:=1 to n do

begin

for j:=1 to m do

begin

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

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

end;

end;

blockwrite(tf,summ\_otr\_bin,sizeof(integer));

closefile(tf);

end;

exports summ\_otr\_bin;

begin

end.

На Рисунках 1 - 8 представлены результаты работы программы:

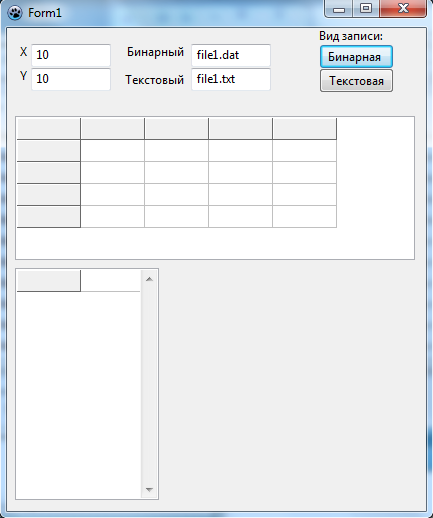


Рисунок 1 - Начальное состояние программы

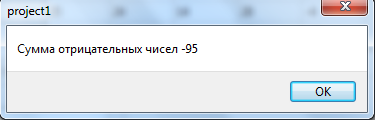


Рисунок 2 - Промежуточный результат работы в бинарном режиме

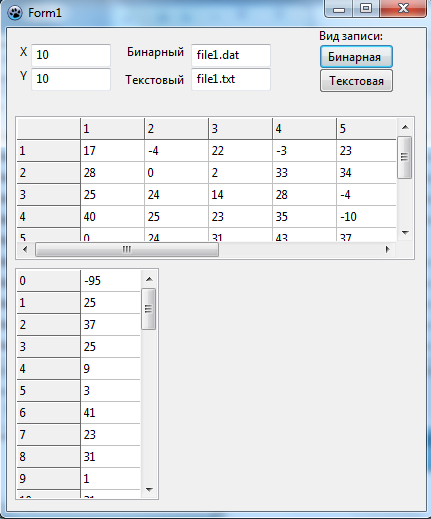


Рисунок 3 - Результат работы программы в бинарном режиме



Рисунок 4 - Созданный .dat файл

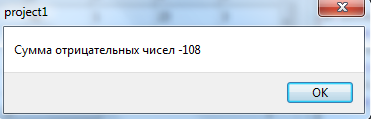


Рисунок 5 - Промежуточный результат работы в текстовом режиме

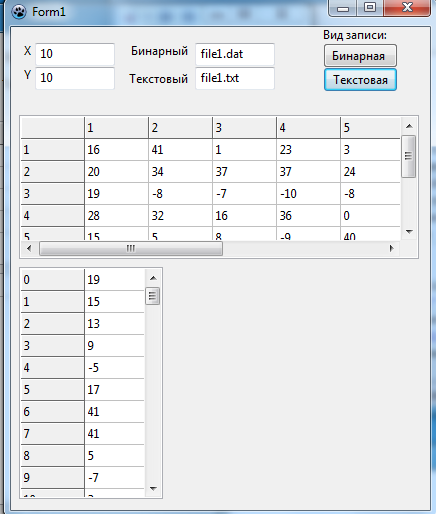


Рисунок 6 - Результат работы программы в текстовом режиме



Рисунок 7 - Созданный .txt файл

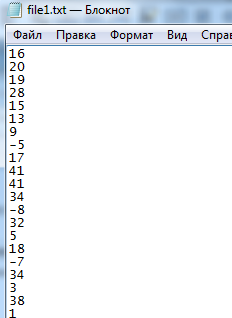


Рисунок 8 - Наполнение .txt файла