**Пример консольного приложения   
с вводом чисел для бинарного дерева из текстового файла   
и выводом его (не более 5 ярусов) на экран**

Пример текстового файла Tree.txt

10

20

30

\*\*

\*\*

40

\*\*

\*\*

50

\*\*

60

70

-90

\*\*

\*\*

100

\*\*

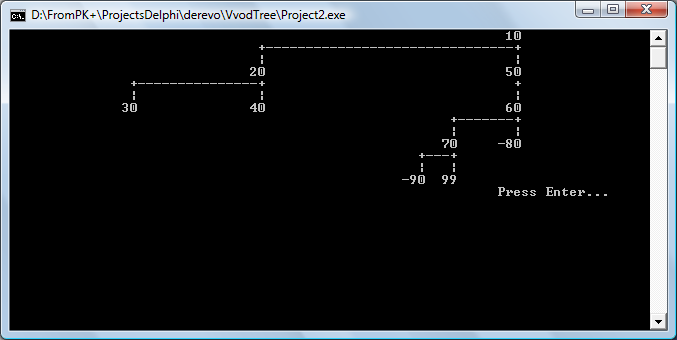
\*\*

-80

\*\*

\*\*

Результат



Добавлена корректировка:

Если число меньше -99, то оно исправляется на -99.

Если число больше 99, то оно исправляется на 99, например, 100 в рассматриваемом примере.

Аномалии не рассмотрены.

Программный код на Delphi

program Project2;

{$APPTYPE CONSOLE}

uses SysUtils;

type

Pder = ^Tder;

Tder = record // моделирование на основе двусвязного нелинейного списка

chislo: integer;

left, rigth: Pder;

end;

var

derevo: array [3..15] of string; // изображение дерева, не более 5 ярусов

procedure create\_tree(var root: Pder; var f: TextFile; i:byte; var kol:byte);

var r: integer; str: string[3];

begin

readln(f, str);

if TryStrToInt(str, r) then

begin

if r>99 then r:=99;

if r<-99 then r:=-99;

if i>kol then kol:=i;

new(root);

root^.chislo:=r; {-99..+99}

create\_tree(root^.left, f, i+1, kol);

create\_tree(root^.rigth, f, i+1, kol);

end

else

root:=nil;

end;

procedure vert\_derevo(var root: Pder; kol: byte=0);

begin

if root <> Nil then

begin

if kol= 0 then writeln('Derevo:'#13#10, root^.chislo)

else writeln('-----':(kol\*5), root^.chislo);

if root^.left<> root^.rigth then // хоть одно есть

begin

vert\_derevo(root^.rigth, kol+1);

vert\_derevo(root^.left, kol+1);

end

end

else writeln('-----':(kol\*5));

end;

procedure free\_tree(var root: Pder);

begin

if root<> nil then

begin

free\_tree(root^.left);

free\_tree(root^.rigth);

dispose(root);

root:=nil;

end;

end;

procedure make\_tree(root: Pder; i,k,kol: byte; r: boolean);

var j:byte;

begin

if i<=kol then

if root <> nil then

begin

if i>1 then

begin

if r then

for j:=1 to 64 div k -1 do derevo[3\*i-2]:=derevo[3\*i-2]+'-'

else

for j:=1 to 64 div k -1 do derevo[3\*i-2]:=derevo[3\*i-2]+' ';

derevo[3\*i-2]:=derevo[3\*i-2]+'+';

for j:=1 to 64 div k -1 do derevo[3\*i-1]:=derevo[3\*i-1]+' ';

derevo[3\*i-1]:=derevo[3\*i-1]+'|';

end;

for j:=1 to 64 div k - 3 do derevo[3\*i]:=derevo[3\*i]+' ';

//знак

if root^.chislo<0 then derevo[3\*i]:=derevo[3\*i]+'-'

else derevo[3\*i]:=derevo[3\*i]+' ';

//две цифры

derevo[3\*i]:=derevo[3\*i]+Chr(48+abs(root^.chislo) div 10)+ Chr(48+abs(root^.chislo) mod 10);

//аналогично с поддреревьями

make\_tree(root^.left, i+1, 2\*k, kol, false);

make\_tree(root^.rigth, i+1, 2\*k, kol, root^.left<>Nil);

end

else

begin // пустое поддерево - пробелы

if r then

for j:=1 to 64 div k do derevo[3\*i-2]:=derevo[3\*i-2]+'-'

else

for j:=1 to 64 div k do derevo[3\*i-2]:=derevo[3\*i-2]+' ';

for j:=1 to 64 div k do derevo[3\*i-1]:=derevo[3\*i-1]+' ';

for j:=1 to 64 div k do derevo[3\*i]:=derevo[3\*i]+' ';

make\_tree(nil, i+1, 2\*k, kol, false);

make\_tree(nil, i+1, 2\*k, kol, false);

end;

end;

var

root: Pder;

kol,j: byte;

f: TextFile;

begin

AssignFile(f, 'tree.txt');

Reset(f);

kol:=0;

create\_tree(root, f, 1, kol);

CloseFile(f);

if kol>5 then kol:=5;

for j:=3 to kol\*3 do derevo[j]:='';

make\_tree(root, 1, 1, kol, false);

for j:=3 to kol\*3 do writeln(derevo[j]);

write('Press Enter...':75); readln;

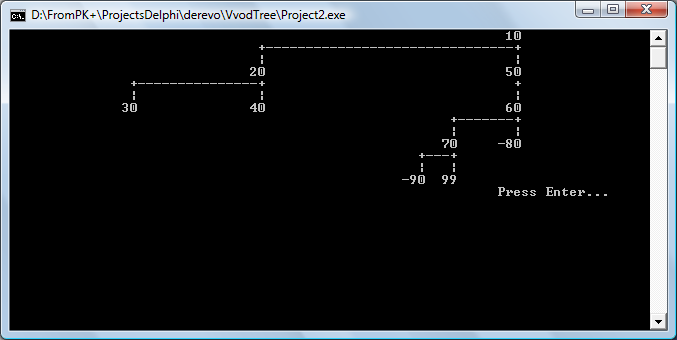
vert\_derevo(root); // вывод дерева вертикально

free\_tree(root);

write('Press Enter...':75); readln;

end.

При выводе в текстовый файл можно изображать дерево графически вертикально или горизонтально, с помощью отступов показывая расположение ветвей дерева:



10

--------- 20

--------- 30

---------\*\*

---------\*\*

--------- 40

---------\*\*

---------\*\*

--------- 50

---------\*\*

--------- 60

--------- 70

--------- -90

---------\*\*

---------\*\*

--------- 99

---------\*\*

---------\*\*

--------- -80

---------\*\*

---------\*\*