Листинг программы

```
unit kalah1;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls,
  ExtCtrls, ComCtrls;

type

  { TForm1 }

  TForm1 = class(TForm)
    Button1: TButton;
    Button10: TButton;
    Button11: TButton;
    Button12: TButton;
    Button13: TButton;
    Button14: TButton;
    Button15: TButton;
    Button16: TButton;
    Button17: TButton;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    Button6: TButton;
    Button7: TButton;
    Button8: TButton;
    Button9: TButton;
    CheckBox1: TCheckBox;
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    TrackBar1: TTrackBar;
    procedure Button15Click(Sender: TObject);
    procedure Button16Click(Sender: TObject);
    procedure Button17Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
  private
    { private declarations }
  public
    { public declarations }
  end;
```

```
type
  position=array [1..14] of integer;

const
  max_depth=8;

var
  Form1: TForm1;
  round,allow_turn,game_over,new_game,comp_game,kalahshot,virtual_kalahshot,
    virtual_capture,capture,overturn,write_in_file:boolean;
  n,best_turn,recurs,depth,rounds,hollow,captured_stones:integer;
  mark_turn:array [8..13] of integer;
  a:position;
  log_file:text;
  game_name:string;
  date:TDateTime;

implementation

{$R *.lfm}

{ TForm1 }

procedure writing_rotate;
begin
  assign(log_file,game_name);
  append(log_file);
  write(log_file,timetostr(now),' доска перевёрнута');
  writeln(log_file,'');
  close(log_file);
end;

function rotate (b:position):position;
var
  i:integer;
  b1:position;
begin
  for i:=1 to 14 do
    If i<=7 then b1[i]:=b[i+7] else b1[i]:=b[i-7];
  rotate:=b1;
end;
```

```
function evaluate_position (b:position):integer;
var
  i:integer;
begin
  evaluate_position:=(b[7]-b[14])*100+(b[1]+b[2]+b[3]+b[4]+b[5]+b[6])-
  (b[8]+b[9]+b[10]+b[11]+b[12]+b[13]);
  // Чем больше значение, тем хуже для компьютера
  If b[7]>=36 then evaluate_position:=3599;
  If b[14]>=36 then evaluate_position:=-3599;
  If  ((b[1]=0)and(b[2]=0)and(b[3]=0)and(b[4]=0)and(b[5]=0)and(b[6]=0))or
  ((b[8]=0)and(b[9]=0)and(b[10]=0)and(b[11]=0)and(b[12]=0)and(b[13]=0))  then
    begin
      for i:=1 to 6 do
        begin
          b[7]:=b[7]+b[i];
          b[14]:=b[14]+b[i+7];
        end;
      If b[7]>b[14]  then evaluate_position:=3599;
      If b[14]>b[7]  then evaluate_position:=-3599;
      If b[7]=b[14]  then evaluate_position:=0;
    end;
end;
```

```
.
.  procedure writing_in_file;
.  var
120    i:integer;
.  begin
.    assign(log_file,game_name);
.    If not(new_game) then
.      begin
125      append(log_file);
.        write(log_file,timetostr(now),' раунд №',rounds,' сложность=',depth);
.        for i:=1 to 14 do
.          If (i=7) or (i=14) then write(log_file,' #',a[i])
.          else write(log_file,' ',a[i]);
130      write(log_file,' &',evaluate_position(a));
.        If not(round) then write(log_file,' игрок походил из лунки ',hollow)
.        else write(log_file,' компьютер походил из лунки ',hollow);
.        If kalahshot then write(log_file,' и точно попал в калах');
.        If capture then write(log_file,' и захватил ',captured_stones,' камней');
135      If round then write(log_file,',');
.        If round then
.          for i:=8 to 13 do
.            write(log_file,' ',mark_turn[i]);
.        writeln(log_file,'');
140      close(log_file);
.      end
.    else
.      begin
.        rewrite(log_file);
145      new_game:=false;
.        write(log_file,Form1.Caption);
.        If comp_game then writeln(log_file,', Болван vs Компьютер')
.        else writeln(log_file,', Игрок vs Компьютер');
.        write(log_file,timetostr(now),' раунд №',rounds,' сложность=0');
150      for i:=1 to 14 do
.          If (i=7) or (i=14) then write(log_file,' #',a[i])
.          else write(log_file,' ',a[i]);
.        writeln(log_file,'');
.        close(log_file);
155    end;
.    If game_over then
.      begin
.        append(log_file);
.        If a[7]>a[14] then write(log_file,timetostr(now),' Игрок выиграл!');
160      If a[14]>a[7] then write(log_file,timetostr(now),' Компьютер выиграл!');
.        If a[7]=a[14] then write(log_file,timetostr(now),' Ничья');
.        close(log_file);
.      end;
.  end;
165
```

```
165
  procedure turn (x:integer);
  var
    i,y:integer;
  begin
170   kalahshot:=false;
    capture:=false;
    captured_stones:=0;
    If not(game_over) then
      begin
175     n:=a[x];
        i:=1;
        a[x]:=0;
        If round then
          begin
180         If n+x<15 then y:=(n+x) mod 14 else
              If n+x<28 then y:=(n+x+1) mod 14 else y:=(n+x+2) mod 14;
            If y=0 then y:=14;
            If y=7 then kalahshot:=true;
            If (n=13) or ((a[y]=0) and (a[14-y]<>0) and (y<7)) then capture:=true;
185         while i<>(n+1) do
              begin
                If ((i+x)<>14)and((i+x)<>28) then
                  If i>(14-x) then
                    If i>(28-x) then
190                   a[i-(28-x)]:=a[i-(28-x)]+1
                    else a[i-(14-x)]:=a[i-(14-x)]+1
                  else a[i+x]:=a[i+x]+1
                else n:=n+1;
                i:=i+1;
195           end;
            If capture then
              begin
                a[7]:=a[7]+a[y]+a[14-y];
                captured_stones:=a[y]+a[14-y];
200             a[y]:=0;
                a[14-y]:=0;
              end;
          end
        else
```

```
205         begin
              If n+x<21 then y:=(n+x) mod 14 else
                If n+x<34 then y:=(n+x+1) mod 14 else y:=(n+x+2) mod 14;
              If y=0 then kalahshot:=true;
              If y=0 then y:=14;
210           If (n=13) or ((a[y]=0) and (a[14-y]<>0) and (y>7)) then capture:=true;
              while i<>(n+1) do
                begin
                  If ((i+x)<>21)and((i+x)<>35) then
                    If i>(14-x) then
215                   If i>(28-x) then
                        a[i-(28-x)]:=a[i-(28-x)]+1
                      else a[i-(14-x)]:=a[i-(14-x)]+1
                    else a[i+x]:=a[i+x]+1
                  else n:=n+1;
220               i:=i+1;
                end;
              If capture then
                begin
                  a[14]:=a[14]+a[y]+a[14-y];
225               captured_stones:=a[y]+a[14-y];
                  a[y]:=0;
                  a[14-y]:=0;
                end;
            end;
230       round:=not(round);
        end;
      rounds:=rounds+1;
    end;
```

```
.
.  procedure turn_checking (x:integer);
.  begin
.    allow_turn:=false;
230  If not(game_over) then
.      begin
.        If (x>0)and(x<7) then If round then If a[x]>0 then allow_turn:=true;
.        If (x>7)and(x<14) then If not(round) then If a[x]>0 then allow_turn:=true;
.      end;
235  end;
.
.  procedure win_checking;
.  var
.    i:integer;
240  begin
.    If ((a[1]=0)and(a[2]=0)and(a[3]=0)and(a[4]=0)and(a[5]=0)and(a[6]=0))or
.    ((a[8]=0)and(a[9]=0)and(a[10]=0)and(a[11]=0)and(a[12]=0)and(a[13]=0)) then
.      begin
.        for i:=1 to 6 do
245        begin
.            a[7]:=a[7]+a[i];
.            a[14]:=a[14]+a[i+7];
.            a[i]:=0;
.            a[i+7]:=0;
250        end;
.        game_over:=true;
.      end
.    else
.      begin
255      If (a[7]>36)or(a[14]>36) then game_over:=true;
.      end;
.  end;
.
.  function virtual_win_checking(b:position):boolean;
260  begin
.    virtual_win_checking:=false;
.    If b[7]>36 then virtual_win_checking:=true;
.    If b[14]>36 then virtual_win_checking:=true;
.    If ((b[1]=0)and(b[2]=0)and(b[3]=0)and(b[4]=0)and(b[5]=0)and(b[6]=0))or
265  ((b[8]=0)and(b[9]=0)and(b[10]=0)and(b[11]=0)and(b[12]=0)and(b[13]=0))
.    then virtual_win_checking:=true;
.  end;
.
```

```
285
.  function virtual_turn (b1:position;i1:integer):position;
.  var
.    n1,k1,y:integer;
.  begin
290  virtual_kalahshot:=false;
.    virtual_capture:=false;
.    n1:=b1[i1];
.    k1:=1;
.    b1[i1]:=0;
295  If n1+i1<21 then y:=(n1+i1) mod 14 else
.      If n1+i1<34 then y:=(n1+i1+1) mod 14 else y:=(n1+i1+2) mod 14;
.    If y=0 then virtual_kalahshot:=true;
.    If y=0 then y:=14;
.    If (n1=13) or ((b1[y]=0)and(b1[14-y]<>0)and(y>7)) then virtual_capture:=true;
300  while k1<(n1+1) do
.      begin
.        If ((k1+i1)<>7)and((k1+i1)<>21)and((k1+i1)<>35) then
.          If k1>(14-i1) then
.            If k1>(28-i1) then
305           b1[k1-(28-i1)]:=b1[k1-(28-i1)]+1
.            else b1[k1-(14-i1)]:=b1[k1-(14-i1)]+1
.          else b1[k1+i1]:=b1[k1+i1]+1
.        else n1:=n1+1;
.        k1:=k1+1;
310    end;
.    If virtual_capture then
.      begin
.        b1[14]:=b1[14]+b1[y]+b1[14-y];
.        b1[y]:=0;
315      b1[14-y]:=0;
.      end;
.    virtual_turn:=b1;
.  end;
.
```

```
.
320 function analysis (b:position;round_counter:integer;player:boolean):integer;
. var
.   i1,mark_turn1,local_recurs:integer;
.   b1:position;
. begin
325   mark_turn1:=-3600;
.   for i1:=13 downto 8 do
.     begin
.       b1:=b;
.       If (b1[i1]=0)and(round_counter=depth) then mark_turn[i1]:=0;
330       If b1[i1]>0 then
.         begin
.           b1:=virtual_turn(b1,i1);
.           If not(virtual_kalahshot) then
.             begin
335             b1:=rotate(b1);
.               If (round_counter>1)and(not(virtual_win_checking(b1))) then
.                 begin
.                   local_recurs:=-analysis(b1,round_counter-1,not(player));
.                   If round_counter=depth then mark_turn[i1]:=local_recurs;
340                 If (player) and (mark_turn1<local_recurs) then
.                     begin
.                       mark_turn1:=local_recurs;
.                       If round_counter=depth then best_turn:=i1;
.                     end;
345                 If (not(player)) and (mark_turn1<local_recurs)
.                   then mark_turn1:=local_recurs;
.                 end;
.               If (round_counter>1)and(virtual_win_checking(b1)) then
.                 begin
350                 If round_counter=depth then mark_turn[i1]:=evaluate_position(b1);
.                   If (player) and (mark_turn1<evaluate_position(b1)) then
.                     begin
.                       mark_turn1:=evaluate_position(b1);
.                       If round_counter=depth then best_turn:=i1;
355                   end;
.                   If (not(player)) and (mark_turn1<evaluate_position(b1))
.                   then mark_turn1:=evaluate_position(b1);
.                 end;
.               If (round_counter=1)and(player)and(mark_turn1<evaluate_position(b1))
360             then mark_turn1:=evaluate_position(b1);
.               If (round_counter=1)and(not(player))and(mark_turn1<evaluate_position(b1))
.               then mark_turn1:=evaluate_position(b1);
.             end
.           else
```

```
            begin
              If (round_counter>1)and(not(virtual_win_checking(b1))) then
                begin
                  local_recurs:=analysis(b1,round_counter-1,player);
                  If round_counter=depth then mark_turn[i1]:=local_recurs;
                  If (player) and (mark_turn1<local_recurs) then
                    begin
                      mark_turn1:=local_recurs;
                      If round_counter=depth then best_turn:=i1;
                    end;
                  If (not(player)) and (mark_turn1<local_recurs)
                  then mark_turn1:=local_recurs;
                end;
              If (round_counter>1)and(virtual_win_checking(b1)) then
                begin
                  If round_counter=depth then mark_turn[i1]:=evaluate_position(b1);
                  If (player) and (mark_turn1<evaluate_position(b1)) then
                    begin
                      mark_turn1:=evaluate_position(b1);
                      If round_counter=depth then best_turn:=i1;
                    end;
                  If (not(player)) and (mark_turn1<evaluate_position(b1))
                  then mark_turn1:=evaluate_position(b1);
                end;
              If (round_counter=1)and(player)and(mark_turn1<evaluate_position(b1))
              then mark_turn1:=evaluate_position(b1);
              If (round_counter=1)and(not(player))and(mark_turn1<evaluate_position(b1))
              then mark_turn1:=evaluate_position(b1);
            end;
        end;
    end;
  analysis:=mark_turn1;
end;
```

```
.
.  procedure refresh_captions;
400  begin
.    Form1.Button1.Caption:=inttostr(a[1]);
.    Form1.Button2.Caption:=inttostr(a[2]);
.    Form1.Button3.Caption:=inttostr(a[3]);
.    Form1.Button4.Caption:=inttostr(a[4]);
405  Form1.Button5.Caption:=inttostr(a[5]);
.    Form1.Button6.Caption:=inttostr(a[6]);
.    Form1.Button7.Caption:=inttostr(a[7]);
.    Form1.Button8.Caption:=inttostr(a[8]);
.    Form1.Button9.Caption:=inttostr(a[9]);
410  Form1.Button10.Caption:=inttostr(a[10]);
.    Form1.Button11.Caption:=inttostr(a[11]);
.    Form1.Button12.Caption:=inttostr(a[12]);
.    Form1.Button13.Caption:=inttostr(a[13]);
.    Form1.Button14.Caption:=inttostr(a[14]);
415  If not(game_over)and(not(new_game)) then
.      begin
.        If not(round) then Form1.label1.Caption:=('Игрок походил из лунки '
.        +inttostr(hollow)+#10#13)
.        else Form1.label1.Caption:=('Компьютер походил из лунки '
420      +inttostr(hollow)+#10#13);
.        If kalahshot then Form1.label1.Caption:=Form1.label1.Caption
.        +(' и точно попал в калах');
.        If capture then Form1.label1.Caption:=Form1.label1.Caption
.        +(' и захватил '+inttostr(captured_stones)+' камней');
425    end;
.    If new_game then
.      begin
.        Form1.label1.Caption:=('Ход игрока');
.        If not(write_in_file) then new_game:=false;
430    end;
.    If game_over then
.      begin
.        Form1.label1.Refresh;
.        sleep(700);
435      If a[7]>a[14] then Form1.label1.Caption:='Игрок выиграл!';
.        If a[14]>a[7] then Form1.label1.Caption:='Компьютер выиграл!';
.        If a[7]=a[14] then Form1.label1.caption:='Ничья';
.      end;
.    Form1.label1.Refresh;
440  sleep(700);
.  end;
.
```

```
445 procedure button_click (x:integer);
    var
      i:integer;
    begin
      depth:=0;
450   turn_checking(x);
      If allow_turn then
      begin
        hollow:=x;
        turn(x);
455     win_checking;
        If game_over then Form1.CheckBox1.Enabled:=true;
        If write_in_file then writing_in_file;
        refresh_captions;
        If kalahshot or game_over then round:=true else
460       begin
            repeat
              If Form1.TrackBar1.Position=1 then
                begin
                  for i:=8 to 13 do
465                 If a[i]>0 then best_turn:=i;
                end;
              If Form1.TrackBar1.Position>1 then
                begin
                  If Form1.TrackBar1.Position=2 then depth:=2;
470               If Form1.TrackBar1.Position=3 then depth:=max_depth;
                  for i:=8 to 13 do
                   If a[i]=0 then depth:=depth+1;
                  If depth>max_depth then depth:=max_depth;
                  analysis(a,depth,true);
475             end;
              hollow:=best_turn;
              turn(best_turn);
              win_checking;
              If game_over then Form1.CheckBox1.Enabled:=true;
480           If write_in_file then writing_in_file;
              refresh_captions;
              If kalahshot then round:=false;
            until not(kalahshot);
          end;
485     If comp_game then
          for i:=6 downto 1 do
            If a[i]>0 then button_click(i);
      end;
    end;
490
```

```
490
.   procedure TForm1.Button1Click(Sender: TObject);
.   begin
.     button_click(1);
.   end;
495
.   procedure TForm1.Button2Click(Sender: TObject);
.   begin
.     button_click(2);
.   end;
500
.   procedure TForm1.Button3Click(Sender: TObject);
.   begin
.     button_click(3);
.   end;
505
.   procedure TForm1.Button4Click(Sender: TObject);
.   begin
.     button_click(4);;
.   end;
510
.   procedure TForm1.Button5Click(Sender: TObject);
.   begin
.    button_click(5);
.   end;
515
.   procedure TForm1.Button6Click(Sender: TObject);
.   begin
.     button_click(6);
.   end;
520
.   procedure TForm1.Button15Click(Sender: TObject);
.   var
.     i:integer;
.   begin
525   write_in_file:=Form1.CheckBox1.Checked;
.     for i:=1 to 14 do
.       if ((i=7) or (i=14)) then a[i]:=0 else a[i]:=6;
.     game_over:=false;
.     rounds:=0;
530   round:=true;
.     comp_game:=false;
.     new_game:=true;
.     refresh_captions;
.     date:=now;
535   game_name:='kalah_'+datetostr(date)+' '+timetostr(date)+'.txt';
.     for i:=1 to 100 do
.       If game_name[i]=':' then game_name[i]:='-';
.     If write_in_file then writing_in_file;
.     Form1.CheckBox1.Enabled:=false;
540 end;
```

```
.
.  procedure TForm1.Button16Click(Sender: TObject);
.  begin
.    a:=rotate(a);
545  If not(new_game) then refresh_captions;
.    If write_in_file then writing_rotate;
.  end;
.
.  procedure TForm1.Button17Click(Sender: TObject);
550  var
.    i:integer;
.  begin
.    write_in_file:=Form1.CheckBox1.Checked;
.    for i:=1 to 14 do
555    if ((i=7) or (i=14)) then a[i]:=0 else a[i]:=6;
.    game_over:=false;
.    rounds:=0;
.    round:=true;
.    comp_game:=true;
560  new_game:=true;
.    refresh_captions;
.    date:=now;
.    game_name:='kalah_'+datetostr(date)+' '+timetostr(date)+'.txt';
.    for i:=1 to 100 do
565    If game_name[i]=':' then game_name[i]:='-';
.    If write_in_file then writing_in_file;
.    Form1.CheckBox1.Enabled:=false;
.    button_click(1);
.  end;
570
.  begin
.    recurs:=1;
.    game_over:=false;
.    new_game:=true;
575  end.
```