

Érdekes informatika feladatok

XXXV. rész

A bináris fa mint graftál kirajzolása

Az elméleti részben bevezettük a *graftál* fogalmát. Nézzük meg most a gyakorlatban, hogyan is történik egy *bináris fa* kirajzolása graftálként. A bináris fa minden egyes ágából újabb két ág ágazik ki 45°-os szögben jobbra, illetve balra. Az utolsó szinten lévő ágakat *leveleknek* nevezzük.

Ha graftálként értelmezzük a bináris fát, akkor a következő leírónyelvet és szabályrendszert adhatjuk meg:

- axióma:
 - o 0
- szabályok:
 - o $1 \rightarrow 11$
 - o $0 \rightarrow 1[0]0$
 - o $[\rightarrow [$
 - o $] \rightarrow]$

Vagyis a nyelv a 0, 1, [,] jelekből áll, a szabályok pedig a következők: a következő iterációs szinten minden 1-est le kell cserélni 11-re, minden 0-ást pedig 1[0]0-re.

Így tehát a különböző iterációs szintek a következőképpen alakulnak:

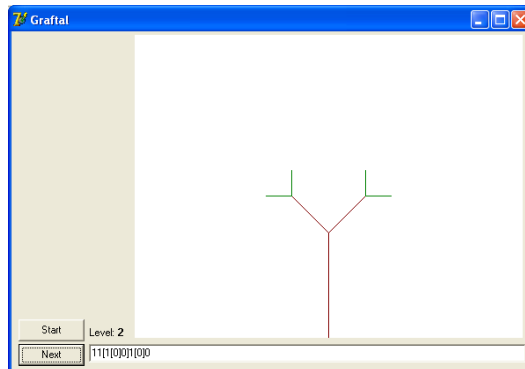
- Axióma, 0-ás szint: 0
- Első szint: 1[0]0
- Második szint: 11[1[0]0]1[0]0
- Harmadik szint: 1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0
- Negyedik szint: 11111111[1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0]1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0
- stb.

Az egyes jelek a következő lépéseket jelölik:

- 0: Rajzolj egy levelet
- 1: Rajzolj egy ágat
- [: Mentsd le a verembe a pozíciót és a szöveget, majd fordulj 45°-kal balra
-]: Vedd ki a veremből a pozíciót és a szöveget, majd fordulj 45°-kal jobbra

A fentieket összesítve, kiegészítve a veremkezeléssel és a forgatás képleteinek leprogramozásával, megírhatjuk azt a *Borland Delphi* programot, amely graftálként kirajzol egy bináris fát.

Az alkalmazás képe:



Az alkalmazás fő egysége:

```

unit uMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, StrUtils, ExtCtrls;

type
  TForm1 = class(TForm)
    btnStart: TButton;
    btnNext: TButton;
    edStr: TEdit;
    Label1: TLabel;
    lblLevel: TLabel;
    pbGraf: TPaintBox;
    procedure btnStartClick(Sender: TObject);
    procedure btnNextClick(Sender: TObject);
  end;

var
  Form1: TForm1;

  // iterációs szint
  NrIt: integer = 0;
  // a kezdeti pozíció
  XPos: real = 220;
  YPos: real = 345;
  // kezdőszög
  Angle: real = 0;

  // a verem
  Stack: array of real;
  SP: integer = 0;

implementation

{$R *.dfm}

// levél (az utolsó szinten lévő ág) rajzolása zöld színnel
procedure Draw1;
var
  x, y: real;
  RXPos, RYPos: real;
begin

```

```

Form1.pbGraf.Canvas.Pen.Color := clGreen;
Form1.pbGraf.Canvas.MoveTo(Round(XPos), Round(YPos));
// az új pont
x := XPos;
y := YPos-30;
// az új pont elforgatva az Angle szöggel a régi (XPos, YPos) pont körül
// a szöget átalakítjuk radiánná
RXPos := XPos + ((x-XPos)*cos(Angle*0.0174532925)-(y-
YPos)*sin(Angle*0.0174532925));
RYPos := YPos + ((x-XPos)*sin(Angle*0.0174532925)+(y-
YPos)*cos(Angle*0.0174532925));
XPos := RXPos;
YPos := RYPos;
Form1.pbGraf.Canvas.LineTo(Round(XPos), Round(YPos));
end;

// ág rajzolása barnával
procedure Draw2;
var
  x, y: real;
  RXPos, RYPos: real;
begin
  Form1.pbGraf.Canvas.Pen.Color := clMaroon;
  Form1.pbGraf.Canvas.MoveTo(Round(XPos), Round(YPos));
  // az új pont
  x := XPos;
  y := YPos-60;
  // az új pont elforgatva az Angle szöggel a régi (XPos, YPos) pont körül
  // a szöget átalakítjuk radiánná
  RXPos := XPos + ((x-XPos)*cos(Angle*0.0174532925)-(y-
  YPos)*sin(Angle*0.0174532925));
  RYPos := YPos + ((x-XPos)*sin(Angle*0.0174532925)+(y-
  YPos)*cos(Angle*0.0174532925));
  XPos := RXPos;
  YPos := RYPos;
  Form1.pbGraf.Canvas.LineTo(Round(XPos), Round(YPos));
end;

// a pozíció és a szög elmentése a veremben, forgatás
procedure Push;
begin
  SetLength(Stack, SP+3);
  Stack[SP] := XPos;
  Stack[SP+1] := YPos;
  Stack[SP+2] := Angle;
  Inc(SP, 2);
  Angle := Angle - 45;
end;

// a pozíció és a szög kivétele a veremből, forgatás
procedure Pop;
begin
  Angle := Stack[SP];
  YPos := Stack[SP-1];
  XPos := Stack[SP-2];
  SetLength(Stack, SP-2);
  Dec(SP, 2);
  Angle := Angle + 45;
end;

// a graftál értelmezése és rajzolás a szabályok alapján
procedure Draw(const s: string);
var
  i: integer;
begin
  XPos := 220;
  YPos := 345;

```

```

Angle := 0;
SP := 0;
SetLength(Stack, SP);
Form1.pbGraf.Canvas.Brush.Color := clWhite;
Form1.pbGraf.Canvas.Brush.Style := bsSolid;
Form1.pbGraf.Canvas.FillRect(Form1.pbGraf.ClientRect);
for i := 1 to Length(s) do
  case s[i] of
    '0': Draw1;
    '1': Draw2;
    '[': Push;
    ']': Pop;
  end;
end;

// kezdőbeállítások
procedure TForm1.btnStartClick(Sender: TObject);
begin
  edStr.Text := '0';
  NrIt := 0;
  lblLevel.Caption := '0';
  Draw(edStr.Text);
end;

// a következő iteráció a szabályok alapján
procedure TForm1.btnNextClick(Sender: TObject);
var
  s: string;
begin
  s := edStr.Text;
  s := AnsiReplaceStr(s, '1', '11');
  s := AnsiReplaceStr(s, '0', '1[0]0');
  edStr.Text := s;
  Inc(NrIt);
  lblLevel.Caption := IntToStr(NrIt);
  Draw(edStr.Text);
end;
end.

```

Kovács Lehel István

Az anyagszerkezet megismerésének jelentős pillanatai

110 éve kapott W. C. Röntgen fizikai Nobel-díjat, az X-sugár felfedezéséért

Az ötvenéves W. C. Röntgen, német fizikus 1895-ben egy tíz oldalas közleményben számolt be az általa felfedezett X-sugárzásnak nevezett jelenségről, melynek természetét még nem ismerte, de felhasználhatóságát már sejtette. Ez tette lehetővé a röntgendiagnosztika gyors alkalmazását a humángyógyászatban, valamint a röntgenső-ipar kialakulását és fejlődését. Vizsgálatai eredményeiért 1901-ben az első fizikai Nobel-díjat „a róla elnevezett sugárzás felfedezésével szerzett rendkívüli érdemeiért” indoklással neki ítelték.

