

10 フラクタル図形 2

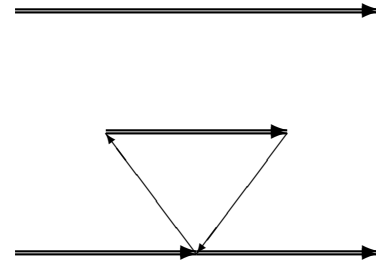
前回, コッホ曲線を紹介しましたが, さらにいくつかフラクタル図形を紹介します。

10.1 ガスケット (Gasket)

10.1.1 作り方

1本の線分を, 長さが半分の平行な線分3本で置き換える。

3本の線分(太線)は連続していないので, 次の線分を描く前にその始点までペンを移動する(細線)。



10.1.2 図形選択

実行時に図形を選べるようにします。

ComboBox をメインパネルの中, レベル指定用パネルの下につけて置く。

Name	ComboBoxZukei
Text	図形選択
Items	コッホ曲線 雪印 ガスケット

描画ボタンの OnClick ハンドラーを変更

```

procedure TFormFractal.ButtonDrawClick(Sender: TObject);
var
  Level : Byte;
begin
  Level := UpDownLevel.Position;
  case ComboBoxZukei.ItemIndex of
    0 : DrawKoch(Level);
    1 : KochThree(Level);
    2 : DrawGasket(Level);
  end;
  UpDownLevel.Position := Level+1; // 自動的にレベルをアップする
end; (* ButtonDrawClick *)

```

10.1.3 メソッド Gasket と DrawGasket

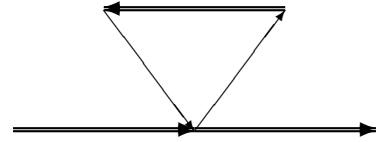
再帰プロシージャ Gasket(Level : Byte; Muki, Nagasa : Real) と, それを使ってガスケットを描く DrawGasket(Level : Byte) を追加しなさい。

作ったプロシージャが正しいかどうか, ホームページに実行プログラムがあるので, それと同じ図形が描かれることを確認してください。

10.2 にせガスケツト

10.2.1 作り方

レベル 1,2 の図形が同じでも、線の向きが異なると、レベル 3 以上の図形は全く違ったものになります。ガスケツトの 2 本目の線を逆向きにすると、どうなるでしょうか。



10.2.2 プログラム

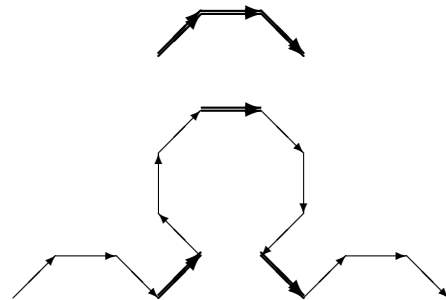
NiseGasket と DrawNiseGasket を追加して実行しなさい。
実行したら、ガスケツトを描いて、消去しないで、にせガスケツトを重ねて描いて見ましょう。

10.3 シェルピンスキー曲線 (Sierpinski Curve)

10.3.1 作り方

- (1) レベル 1 のとき、底辺がない台形を描く。
- (2) レベルが 2 以上のとき、やはり台形の 3 辺 (図の太線) を描くが、台形の 4 つの頂点の部分にレベルが 1 小さい図形 (図の細線) を挿入する。

今までの図形は、実際に線を描くのはレベル 1 だけで、高レベルの図形は低レベルの図形を直接つなげたものになっている。シェルピンスキー曲線の高レベルの図形は低レベルの曲線を 1 本の線を介してつなげている。この仲介線をそれぞれのレベルで描く。



10.3.2 プログラム

再帰プロシージャ Sierpinski(Level : Byte; Muki,Nagasa : Real) と、それを使って曲線を描くプロシージャ DrawSierpinski(Level : Byte) を作りなさい。

プロシージャ Sierpinski で指定する Nagasa は、今までのように始点から終点までの距離ではなく一辺の長さで、最後まで一定です。各レベルで同じ長さの仲介線を描くので、再帰呼び出しの度に段々短くしていくということができないからです。DrawSierpinski で Sierpinski を呼ぶとき、Nagasa を同じ値 (たとえば 10) にすると、上の図のように、レベル 2 の図はレベル 1 の図の 3 倍になってしまいます。レベルを大きくすると図形はどんどん大きくなっていきます。レベルが違って同じ大きさ (始点から終点までの距離が同じ) にするには、Nagasa を Level によって変えないといけません。

問題

- (1) レベルが L の図形はレベル 1 の図形の何倍でしょうか。
- (2) すべてのレベルにおいて、始点から終点までの距離を 600 にするためには Nagasa をどのように定めたらよいでしょうか、Level を用いて表しなさい。

それには n 乗の計算が出てきます。Pascal には “ n 乗” を計算する演算子はありませんが、 $m \times 2^n$ は `M shl N` で計算できます。これは m を 2 進法で表したものを、左に n 桁シフトする (空いた右側には 0 を補う) という意味です。たとえば、13 (2 進法で 00001101) `shl 3` は 104 (2 進法で 01101000) になります。これを使って、`DrawSierpinski` を書いてください。

```

procedure TFormFractal.Sierpinski(Level : Byte; Muki,Nagasa : Real);
    (* シェルピンスキー曲線を描く (再帰) *)
begin
    if Level <= 1
    then begin
        Line(          ,Nagasa);
        Line(          ,Nagasa);
        Line(          ,Nagasa);
    end
    else begin
        Sierpinski(Level-1,          ,Nagasa);
        Line(          ,Nagasa);
        Sierpinski(Level-1,          ,Nagasa);
        Line(          ,Nagasa);
        Sierpinski(Level-1,          ,Nagasa);
        Line(          ,Nagasa);
        Sierpinski(Level-1,          ,Nagasa);
    end;
end; (* Sierpinski *)

procedure TFormFractal.DrawSierpinski(Level : Byte);
    (* シェルピンスキー曲線を描く *)
var
    Nagasa : Real;
begin
    Nagasa :=          ; // 始点から終点までの距離が 600
    StandBy(100,350);
    Sierpinski(Level,0,Nagasa);
end; (* DrawSierpinski *)

```

10.3.3 正方形

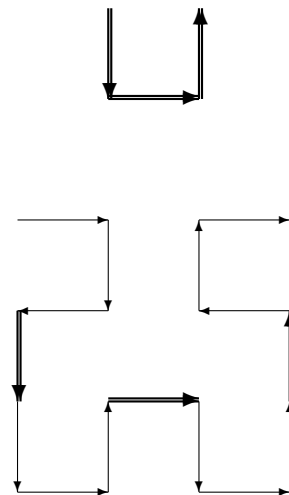
シェルピンスキー曲線を斜めに 2 つ描く (右下に向かって描いて、左上に向かって描く) と正方形形状の図形ができます。ただし、2 つの曲線を伸介線で結ばないと、きれいな正方形になりません。この正方形を描く `SierpinskiTwo(Level : Byte)` を作りなさい。

これは閉曲線なので、内と外があります。外部に色を塗ると面白いでしょう。

10.4 ヒルベルト曲線 (Hilbert Curve)

10.4.1 作り方

- (1) レベル 1 では、正方形の 3 辺を描く。
- (2) レベル 2 以上では、やはり 3 辺 (図の太線) を描くが、4 つの頂点の部分にレベルが 1 低い図形 (細線) を埋め込む。ただし、埋め込む図形を時計回りに描く場合と反時計回りに描く場合がある。



10.4.2 プログラム

図のレベル 1 の 3 辺は、進行方向 (始点から終点に向かう方向) の右側に描いています。すなわち、 $Muki + 90^\circ$ 、 $Muki$ 、 $Muki - 90^\circ$ の順に線を描いています。レベル 2 に埋め込まれた 4 つのレベル 1 の 3 辺は、最初と最後は進行方向の左側に描いています。すなわち、 $Muki - 90^\circ$ 、 $Muki$ 、 $Muki + 90^\circ$ の順に線を描いています。

ゆえに、ヒルベルト曲線を描く再帰プロシージャには、 $Level$ 、 $Muki$ 、 $Nagasa$ の他に $\pm 90^\circ$ を表す引数 $PM90$ を追加するとよいことがわかります。

次のプロシージャを完成させなさい。

```

procedure TFormFractal.Hilbert(Level: Byte; Muki,Nagasa,PM90: Real);
    (* ヒルベルト曲線を描く (再帰) *)
    (* Muki : 自分の向き *)
    (* PM90 : 最初が右折 (+90°) か左折 (-90°) か *)
begin

end; (* Hilbert *)

procedure TFormFractal.DrawHilbert(Level : Byte);
    (* ヒルベルト曲線を描く *)
begin
    Canvas.Pen.Color := clRed;
    StandBy(100,100);
    Hilbert(Level,0,600/(1 shl Level -1),Pi/2);
end; (* DrawHilbert *)

```

これは開曲線なので、外と内と分かれていません。始点まで戻るヒルベルト曲線をもう 1 つ描くと閉曲線になります。そして外側に色を塗りましょう。