

## 4 グラフィックス 星

### 4.1 正 $n : d$ 星

中心が  $C(x_C, y_C)$  , 半径が  $r$  の円周上の点  $P(x, y)$  は偏角  $\theta$  を用いて次のように表されます。

$$\begin{cases} x = x_C + r \cos \theta \\ y = y_C + r \sin \theta \end{cases}$$

この円に内接する正  $n$  角形の頂点  $P_k(x_k, y_k)$  ( $k = 0, 1, \dots, n - 1$ ) は, 次のように表されます。

$$\begin{cases} x_k = x_C + r \cos \theta_k \\ y_k = y_C + r \sin \theta_k \end{cases} \quad \left( \theta_k = \frac{2\pi}{n} \times k \right)$$

ただし, この正  $n$  角形は始点  $P_0(x_0, y_0)$  が  $x$  軸上にあります。角  $\alpha$  だけ回転して始点の偏角が  $\alpha$  になるようにした正  $n$  角形の頂点は, 次のように表されます。

$$\begin{cases} x_k = x_C + r \cos \theta_k \\ y_k = y_C + r \sin \theta_k \end{cases} \quad \left( \theta_k = \alpha + \frac{2\pi}{n} \times k \right)$$

$P_k$  と  $P_{k+1}$  ( $k = 0, 1, \dots, n - 1$ ) を線で結ぶと正  $n$  角形が描かれます<sup>1</sup>。

$P_k$  と  $P_{k+d}$  ( $k = 0, 1, \dots, n - 1$ ) を線で結んでできる図形を正  $n : d$  星と呼ぶことにします。正  $n$  角形は正  $n : 1$  星, いわゆる星形は正  $5 : 2$  星, 六芒星は正  $6 : 2$  星です。

### 4.2 プログラム Star

正  $n : d$  星を描くプログラムを作ります。

#### 4.2.1 新規プログラム登録

フォルダ名	Star
ユニット名	StarU
プロジェクト名	StarP

#### 4.2.2 フォーム変更

Name	FormStar
Caption	正 $n : d$ 星

<sup>1</sup> $P_n, P_{n+1}, \dots$  は  $P_0, P_1, \dots$  と同一視します

### OnCreate ハンドラー

---

```

(***** 一般のプロシージャ *****)
(***** フォームのメソッド *****)
(***** イベントハンドラー *****)
procedure TFormStar.FormCreate(Sender: TObject);
begin
    WindowState := wsMaximized;
end; {FormCreate}

```

---

#### 4.2.3 メインパネル

Panel を FormStar の中に置く。

```

Align      alRight
Name       PanelMain
Caption

```

#### 4.2.4 イメージ

Image をフォームの中、左の方に置く。

```

Align      alClient
Name       ImageStar

```

#### 4.2.5 終了ボタン

Button を PanelMain の中、下の方に置く。

```

Name       ButtonClose
Caption    終了 (&X)
Anchors.akTop    False
Anchors.akBottom True

```

### OnClick ハンドラー

---

```

procedure TFormStar.ButtonCloseClick(Sender: TObject);
begin
    Close;
end; {ButtonCloseClick}

```

---

#### 4.2.6 星形指定用

Panel を PanelMain の中に置く。

```

Align      alTop
Alignment  alLeftJustify
Name       PanelKatachi
Font.Name  MS ゴシック
Caption    星形  n = * * * , d = * * *   は全角空白

```

Edit を 2 つ PanelKatachi の Caption の \* \* \* をちょうど隠すように置く。

Name	EditN	EditD
Text	5	2

#### 4.2.7 中心指定用

PanelKatachi を [編集 | コピー] して, PanelMain の中に [編集 | 貼り付け] する。

Name	PanelChuushin	
Caption	中心 x = * * * , y = * * *	は全角空白
Name	EditXC	EditYC
Text	400	300

#### 4.2.8 始点指定用

もう 1 つ, PanelKatachi を PanelMain の中に貼り付ける。

Name	PanelShiten	
Caption	始点 r = * * * , = * * * °	
Name	EditR	EditAlpha
Text	100	0

#### 4.2.9 色指定用

RadioGroup を PanelMain に追加する。

Align	a1Top
Height	180
Caption	色
Items	暗い黒 暗い赤 暗い緑 暗い黄 暗い青 暗い紫 暗い水 暗い白 明るい黒 明るい赤 明るい緑 明るい黄 明るい青 明るい紫 明るい水 明るい白
Columns	2
ItemIndex	0

## 4.2.10 描画ボタン

Button を PanelMain の中に置く。

```
Name      ButtonDraw
Caption    描画 (&D)
```

OnClick ハンドラー

---

```
procedure TFormStar.ButtonDrawClick(Sender: TObject);
begin
  DataWoYomu(Star);
  HoshiWoEgaku(Star);
end; {ButtonDrawClick }
```

---

## 4.2.11 データ構造

星のデータ構造の定義文を、

```
type
  TFormStar = class(TForm)
```

の間に記入する。

---

```
type
  TIroNo = 0..15;
  TStar = record
    N      : Integer; // 頂点の数
    D      : Integer; // k 番目と k + d 番目を結ぶ
    XC     : Integer; // 中心の x 座標
    YC     : Integer; // 中心の y 座標
    R      : Integer; // 半径
    Alpha  : Integer; // 始点の偏角
    IroNo  : TIroNo; // 16 色の番号
  end;
  TFormStar = class(TForm)
```

---

変数 Star を TFormStar の private 部に宣言する。

---

```
private
  { Private 宣言 }
  Star : TStar;
```

---

## 4.2.12 メソッド追加

2つのプロシージャ,星のデータを読み込む `DataWoYomu` と,星を描く `HoshiWoEgaku` を `TFormStar` のメソッドとして定義する。

private 部にヘッダーを書くことを忘れないように。

---

```
(***** フォームのメソッド *****)
procedure TFormStar.DataWoYomu(var Star : TStar);
    { 星のデータを読む }
begin
    with Star do
        begin
            N := StrToInt(EditN.Text);
            D := StrToInt(EditD.Text);
            XC := StrToInt(EditXC.Text);
            YC := StrToInt(EditYC.Text);
            R := StrToInt(EditR.Text);
            Alpha := StrToInt(EditAlpha.Text);
            IroNo := RadioGroupIro.ItemIndex;
        end;
    end; {DataWoYomu}

procedure TFormStar.HoshiWoEgaku(Star : TStar);
    { 星を描く }
const
    Iro : array [TIroNo] of TColor
        = (clBlack,clMaroon,clGreen,clOlive, clNavy,clPurple, clTeal,clSilver,
           clGray, clRed,    clLime, clYellow,clBlue,clFuchsia,clAqua,clWhite );
var
    Theta : Real;
    K : Byte;
begin
    with ImageStar.Canvas, Star do
        begin
            Pen.Color := Iro[IroNo];
            for K := 1 to N do
                begin
                    Theta := DegreeToRadian(Alpha)+2*Pi/N*K;
                    MoveTo(Round(XC+R*Cos(Theta)),Round(YC-R*Sin(Theta)));
                    Theta := Theta+2*Pi/N*D;
                    LineTo(Round(XC+R*Cos(Theta)),Round(YC-R*Sin(Theta)));
                end;
            end;
        end;
    end; {HoshiWoEgaku}
```

---

## 4.2.13 一般のプロシージャ

問題. 角の単位を変換するファンクション `DegreeToRadian` を記入しなさい。

## 4.2.14 実行

いろいろの正  $n : d$  星を描いてみましょう。

正 5:2 星 (いわゆる星) を 1 つの頂点が真上になるようにするには,  $\alpha$  を何度にしたらよいでしょうか。

## 4.3 改良

### 4.3.1 消去ボタン

Button を PanelMain に追加する。

```
Name      ButtonClear
Caption    消去 (&C)
```

#### OnClick ハンドラー

---

```
procedure TFormStar.ButtonClearClick(Sender: TObject);
begin
  with ImageStar.Canvas do
    begin
      FillRect(ClipRect);
    end;
end; {ButtonClearClick}
```

---

1 行だけですか, FillRect, ClipRect それぞれの前に ImageStar.Canvas. が必要なので, with を使って省略しています。

FillRect は長方形の内部を Brush で塗りつぶす命令で, ClipRect は Canvas 全体の長方形を表します。このプログラムでは, 他で Brush を使っていないので, 単純に 1 行ですみますが, 他で Brush を使っている場合は, Brush.Color や Brush.Style を指定しないとけません。

## 4.4 星を回転させる

始点の偏角  $\alpha$  を少しずつ変えて描くと回転して見えます。

### 4.4.1 回転タイマー

Timer を任意の位置に置く。

```
Name      TimerKaiten
Interval   1
Enabled    False;
```

#### OnTimer ハンドラー

---

```
procedure TFormStar.TimerKaitenTimer(Sender: TObject);
begin
  Kaiten(Star);
end; {TimerKaitenTimer}
```

---

#### 4.4.2 回転/停止ボタン

Button を PanelMain に追加する。  
 Name ButtonKaiten  
 Caption 回転 ( R)  
 OnClick ハンドラー

---

```

procedure TFormStar.ButtonKaitenClick(Sender: TObject);
begin
  TimerKaiten.Enabled := not TimerKaiten.Enabled;
  if TimerKaiten.Enabled
  then ButtonKaiten.Caption := '停止 (&S)'
  else ButtonKaiten.Caption := '回転 (&R)';
end; {ButtonKaitenClick}

```

---

ボタンを押すたびに、タイマーのイベント発生の許可と禁止を切り替えます。すなわち1つのボタンで回転と停止の両方の機能を持っています。ボタンのキャプションを機能に合わせて、「回転」または「停止」と書き換えるようにしてあります。

#### 4.4.3 メソッド追加

---

```

procedure TFormStar.Kaiten(var Star : TStar);
  { 星を回転する }
begin
  with Star do
  begin
    Alpha := Alpha+1; // 偏角を1°ふやす
    HoshiWoEgaku(Star);
  end;
end; {Kaiten}

```

---

$\alpha$  の変更が一時的なものではなく、Star の Alpha を変更しないといけないので、引数の Star の前に var を書いて変数引数にします。

#### 4.4.4 実行

回転しますが、回転前の絵が残っているのでよくありません。

#### 4.4.5 修正

---

```

procedure TFormStar.Kaiten(var Star : TStar);
  { 星を回転する }
var
  IroNoHozon : TIroNo;
begin
  with Star do
  begin

```

```

    IroNoHozon := IroNo;    // 色番号を保存
    IroNo := 15;           // clWhite
    HoshiWoEgaku(Star);    // 消す
    IroNo := IroNoHozon;   // 元に戻す
    Alpha := Alpha+1;      // 偏角を 1° ふやす
    HoshiWoEgaku(Star);
  end;
end; {Kaiten}

```

---

#### 4.4.6 実行

きれいに回転して見えます。

停止して、半分くらい重なるような別の星を描いて回転させてみましょう。

前の星の一部が消えてしまいよくありません。

#### 4.4.7 再修正

```

procedure TFormStar.Kaiten(var Star : TStar);
  { 星を回転する }
begin
  with Star do
    begin
      HoshiWoEgaku(Star);    // 消す
      Alpha := Alpha+1;      // 偏角を 1° ふやす
      HoshiWoEgaku(Star);
    end;
end; {Kaiten}

```

---

もちろん、これだけでは消えません。

HoshiWoEgaku の中に 1 行追加します。

```

procedure TFormStar.HoshiWoEgaku(Star : TStar);
  { 星を描く }
  (* 省略 *)
begin
  with ImageStar.Canvas, Star do
    begin
      Pen.Mode := pmNotXor;          (*_*)
      Pen.Color := Iro[IroNo];
    end;
  end;
end; {HoshiWoEgaku}

```

---

#### 4.4.8 実行

今度は重ねて描いても残っています。

ペンモードについては次回に解説します。



## 4.5 課題

### 4.5.1 回転速度

現在, 回転するとき  $\alpha$  を  $1^\circ$  ずつ増やしていますが, この増分 `PlusAlpha` を `TStar` のプロパティに追加し, 回転ボタンを押したときに  $-5 \sim +5$  の値をランダムに定めるようにしなさい。絶対値が大きいと早く回転します。また正と負では逆向きに回転します。

### 4.5.2 星たち

`private` 部に, 変数 `Stars : array [1..10] of TStar` を追加し, 描画ボタンを押したときに 10 個の星たちのプロパティを (`PlusAlpha` も) ランダムに定め, 回転ボタンを押すと星たちが一斉に回転するようにしなさい。