

Кривая Гильберта

```

unit gilbert_;

interface

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

type
  TForm1 = class(TForm)
    procedure FormPaint(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

```

```

var
  Form1: TForm1;

```

```

implementation

```

```

{$R *.dfm}

```

```

var
  p: integer = 5; // порядок кривой
  u: integer = 7; // длина штриха

```

```

{ Кривая Гильберта состоит из четырех соединенных прямыми элементами: a,b,c и d.
  Каждый элемент строит соответствующая процедура. }

```

```

procedure a(i:integer; canvas: TCanvas); forward;
procedure b(i:integer; canvas: TCanvas); forward;
procedure c(i:integer; canvas: TCanvas); forward;
procedure d(i:integer; canvas: TCanvas); forward;

```

```

// Элементы кривой

```

```

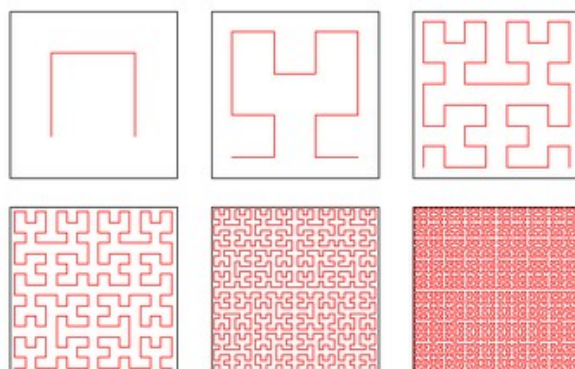
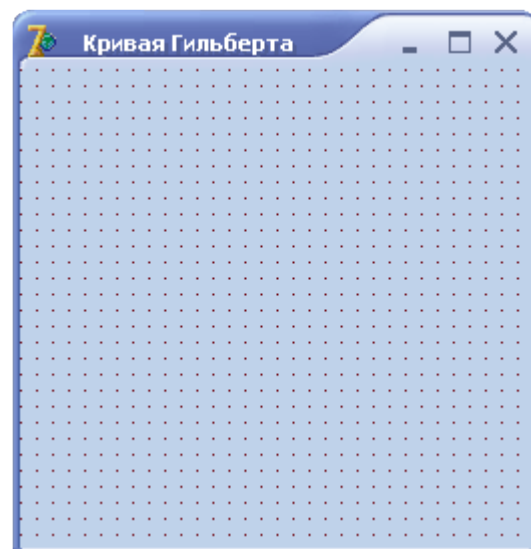
procedure a(i: integer; canvas: TCanvas);
begin
  if i > 0 then begin
    d(i-1, canvas); canvas.LineTo(canvas.PenPos.X+u,canvas.PenPos.Y);
    a(i-1, canvas); canvas.LineTo(canvas.PenPos.X,canvas.PenPos.Y+u);
    a(i-1, canvas); canvas.LineTo(canvas.PenPos.X-u,canvas.PenPos.Y);
    c(i-1, canvas);
  end;
end;

```

```

procedure b(i: integer; canvas: TCanvas);
begin
  if i > 0 then
  begin
    c(i-1, canvas); canvas.LineTo(canvas.PenPos.X-u,canvas.PenPos.Y);

```



```
    b(i-1, canvas); canvas.LineTo(canvas.PenPos.X,canvas.PenPos.Y-u);
    b(i-1, canvas); canvas.LineTo(canvas.PenPos.X+u,canvas.PenPos.Y);
    d(i-1, canvas);
end;
end;
```

```
procedure c(i: integer; canvas: TCanvas);
begin
    if i > 0 then
    begin
        b(i-1, canvas); canvas.LineTo(canvas.PenPos.X,canvas.PenPos.Y-u);
        c(i-1, canvas); canvas.LineTo(canvas.PenPos.X-u,canvas.PenPos.Y);
        c(i-1, canvas); canvas.LineTo(canvas.PenPos.X,canvas.PenPos.Y+u);
        a(i-1, canvas);
    end;
end;
```

```
procedure d(i: integer; canvas: TCanvas);
begin
    if i > 0 then
    begin
        a(i-1, canvas); canvas.LineTo(canvas.PenPos.X,canvas.PenPos.Y+u);
        d(i-1, canvas); canvas.LineTo(canvas.PenPos.X+u,canvas.PenPos.Y);
        d(i-1, canvas); canvas.LineTo(canvas.PenPos.X,canvas.PenPos.Y-u);
        b(i-1, canvas);
    end;
end;
```

procedure TForm1.FormPaint(Sender: TObject);

```
begin
    Form1.Canvas.MoveTo(u,u);
    a(5,Form1.Canvas); // вычертить кривую Гильберта
end;
```

end.