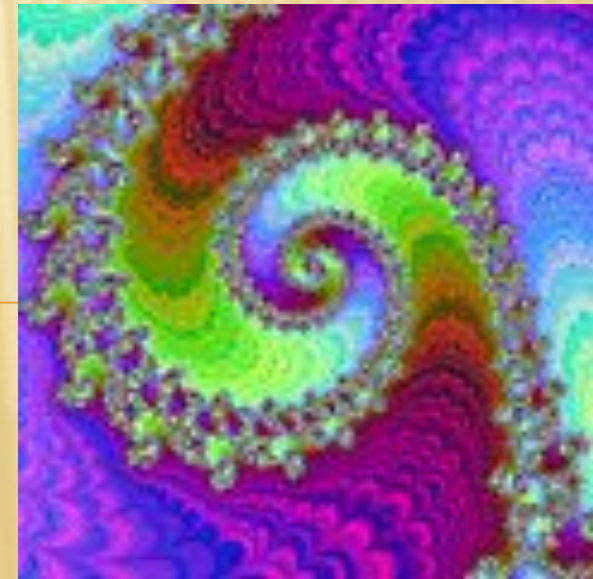
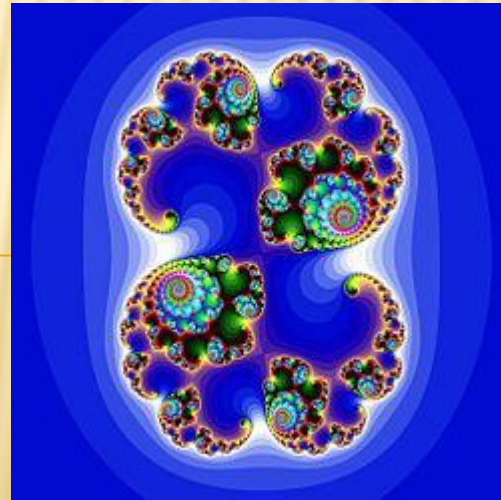
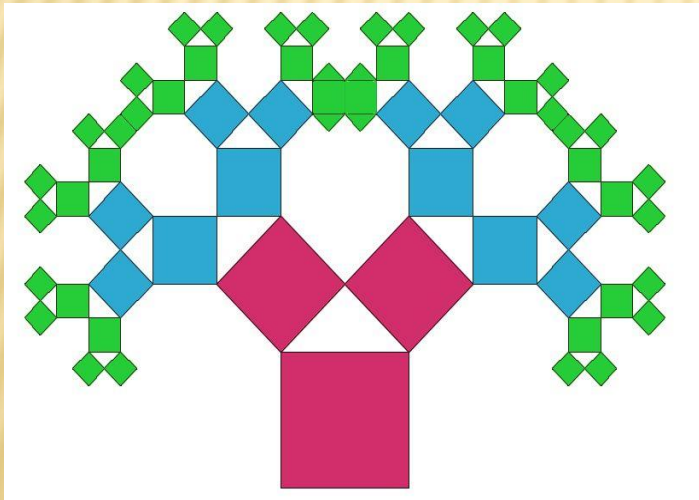
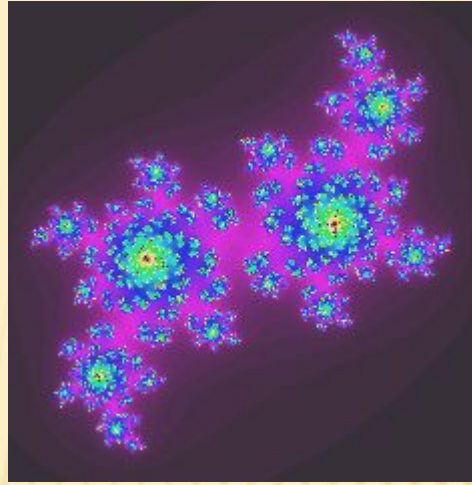
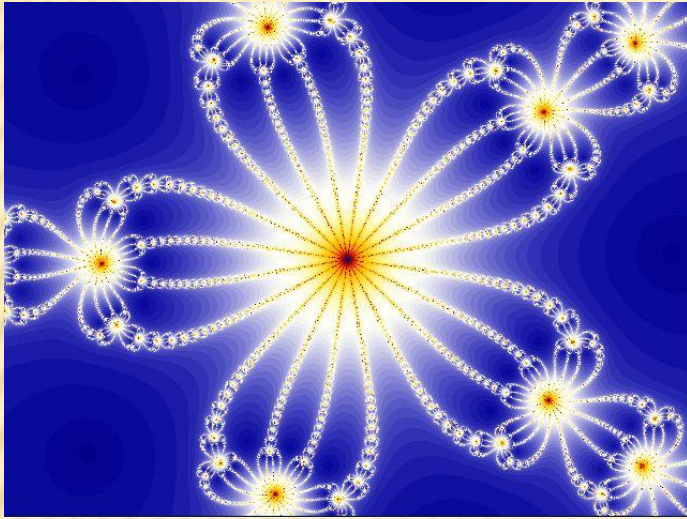


# ФРАКТАЛЬНАЯ ГРАФИКА НА ЯЗЫКЕ TURBO PASCAL

---

**Выполнила :**  
**студентка 3 курса**  
**группы ПО-31**  
**Тупикина Анна**

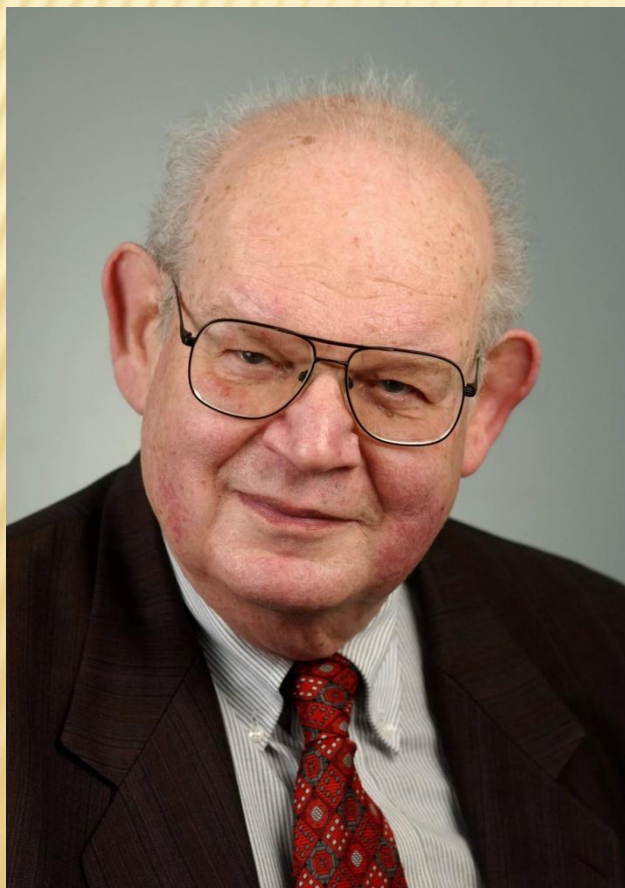
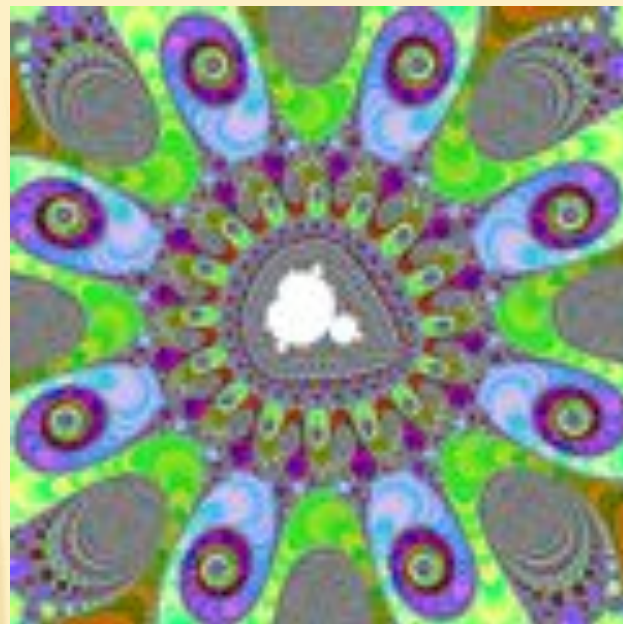
# Фрактал





# ФРАКТАЛ

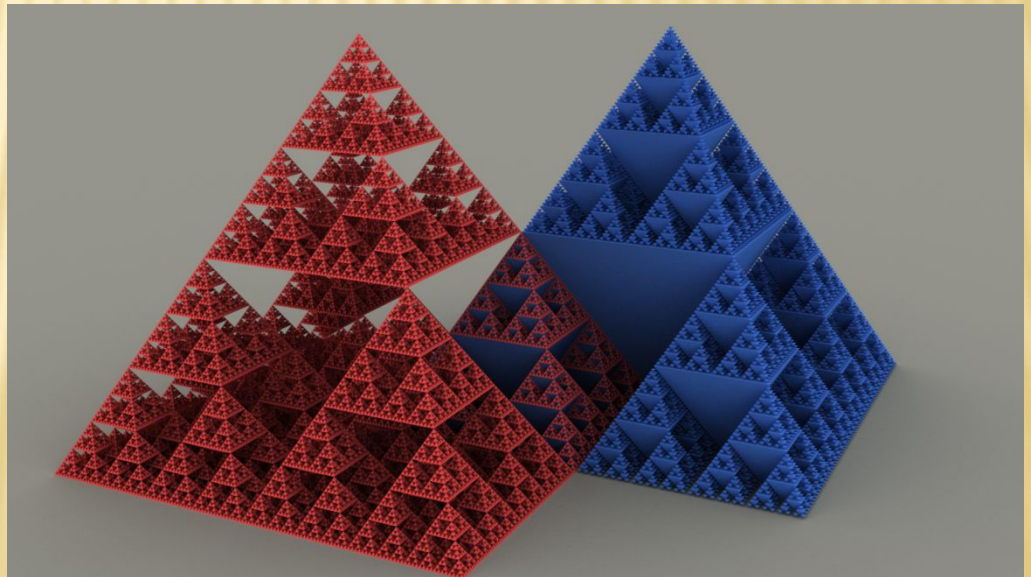
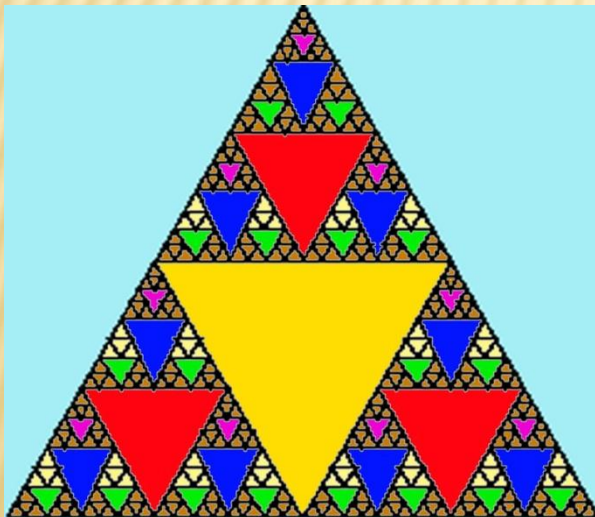
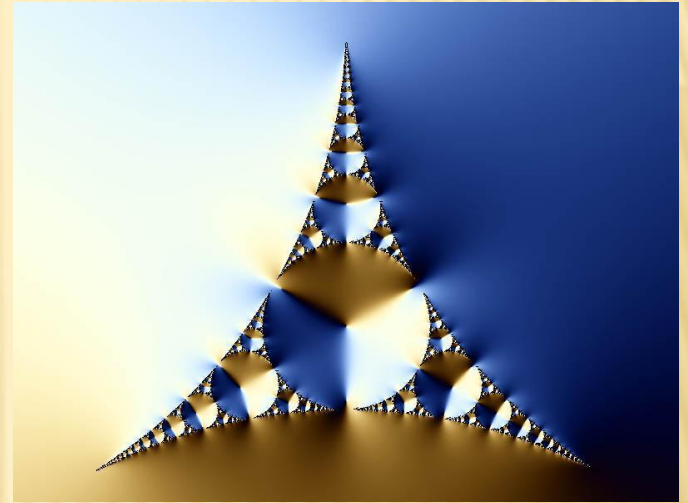
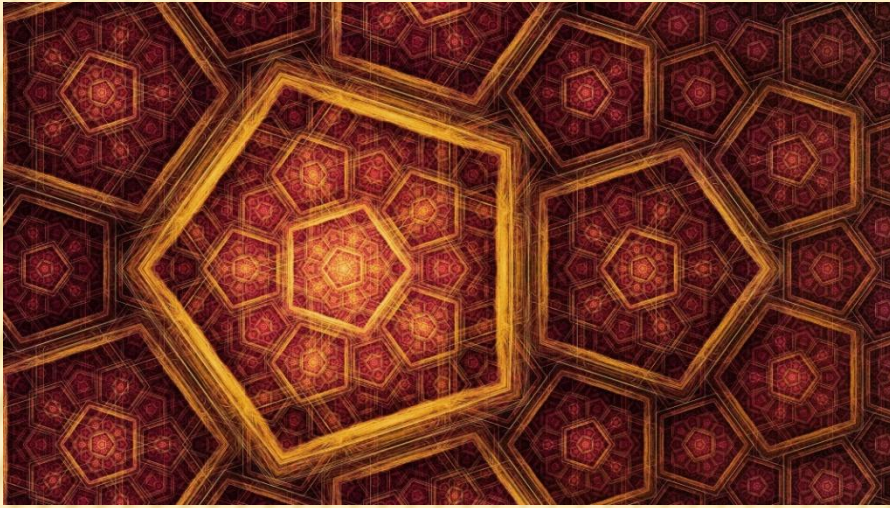
(лат. *fractus* — дроблёный, сломанный, разбитый) — математическое множество, обладающее свойством самоподобия.



Бенуа́ Мандельброт (фр. *Benoît B. Mandelbrot*;) — французский и американский математик, создатель фрактальной геометрии. Лауреат премии Вольфа по физике (1993).



# ФРАКТАЛЬНАЯ ГЕОМЕТРИЯ



# САМОПОДОБИЕ САМОПОДОБНЫЙ ОБЪЕКТ

Это объект, в точности или приближённо совпадающий с частью себя самого (то есть целое имеет ту же форму, что и одна или более частей).

---



# ЯЗЫК ПРОГРАММИРОВАНИЯ ПАСКАЛЬ

Это язык программирования общего назначения. Один из наиболее известных языков программирования, используется для обучения программированию в школах, ССУЗах, ВУЗах, в дальнейшем он служит базой для ряда других языков.

# КЛАССИФИКАЦИЯ ФРАКТАЛОВ

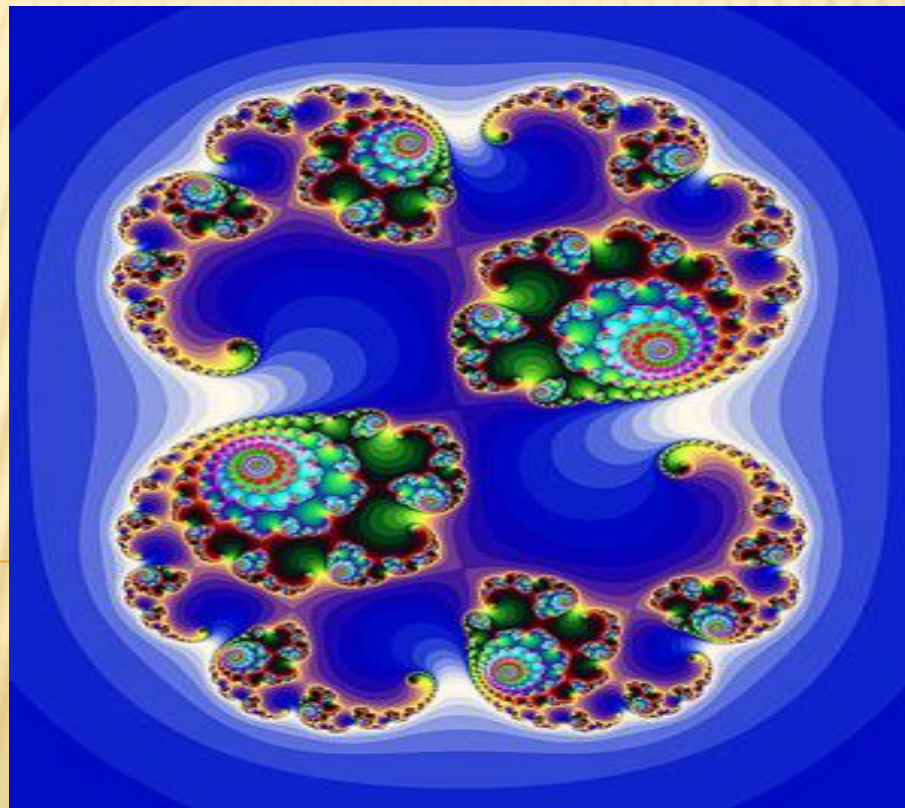
- Обладает нетривиальной структурой на всех масштабах.
  - Является самоподобным или приближенно самоподобным.
  - Обладает дробной метрической размерностью.
-

# В ОСНОВНОМ:

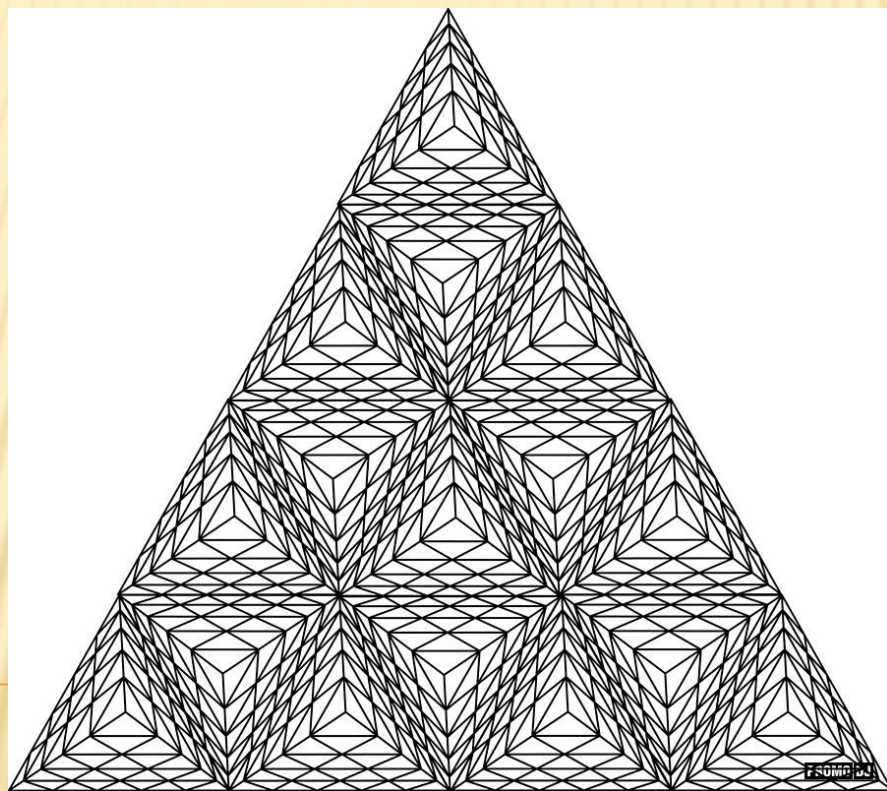
- Алгебраические фракталы .
  - Геометрические фракталы .
  - Стохастические фракталы.
-



**АЛГЕБРАИЧЕСКИЕ ФРАКТАЛЫ - ЭТО САМАЯ КРУПНАЯ ГРУППА ФРАКТАЛОВ, ПОЛУЧИВШАЯ НАЗВАНИЕ ЗА ИСПОЛЬЗОВАНИЕ АЛГЕБРАИЧЕСКИХ ФОРМУЛ.**



**ГЕОМЕТРИЧЕСКИЕ ФРАКТАЛЫ-  
ЭТОТ ТИП ФРАКТАЛОВ ПОЛУЧАЕТСЯ  
ПУТЕМ ПРОСТЫХ ГЕОМЕТРИЧЕСКИХ  
ПОСТРОЕНИЙ.**



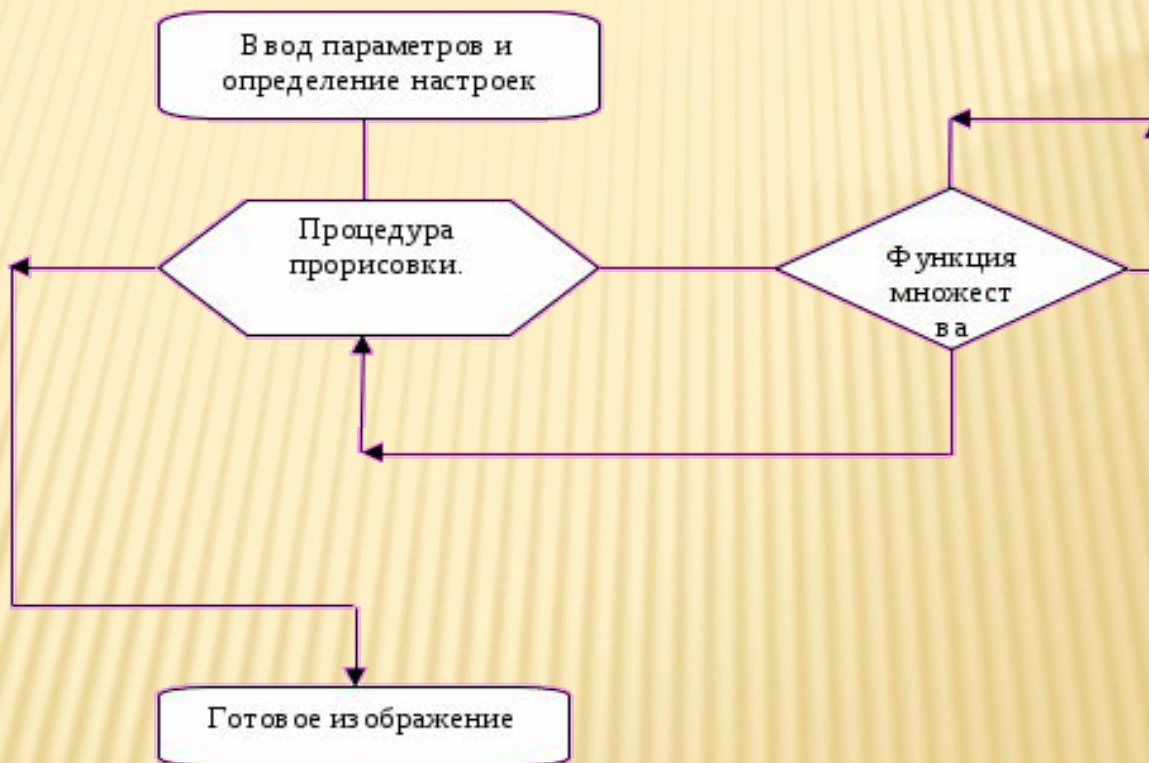


**СТОХАСТИЧЕСКИЕ ФРАКТАЛЫ - ЭТО ВИД ФРАКТАЛОВ, КОТОРЫЙ ПОЛУЧАЮТСЯ В ТОМ СЛУЧАЕ, ЕСЛИ В ИТЕРАЦИОННОМ ПРОЦЕССЕ СЛУЧАЙНЫМ ОБРАЗОМ МЕНЯТЬ КАКИЕ-ЛИБО ЕГО ПАРАМЕТРЫ.**





# СХЕМА РАБОТЫ ПРОГРАММЫ





# ДЕРЕВО ПИФАГОРА

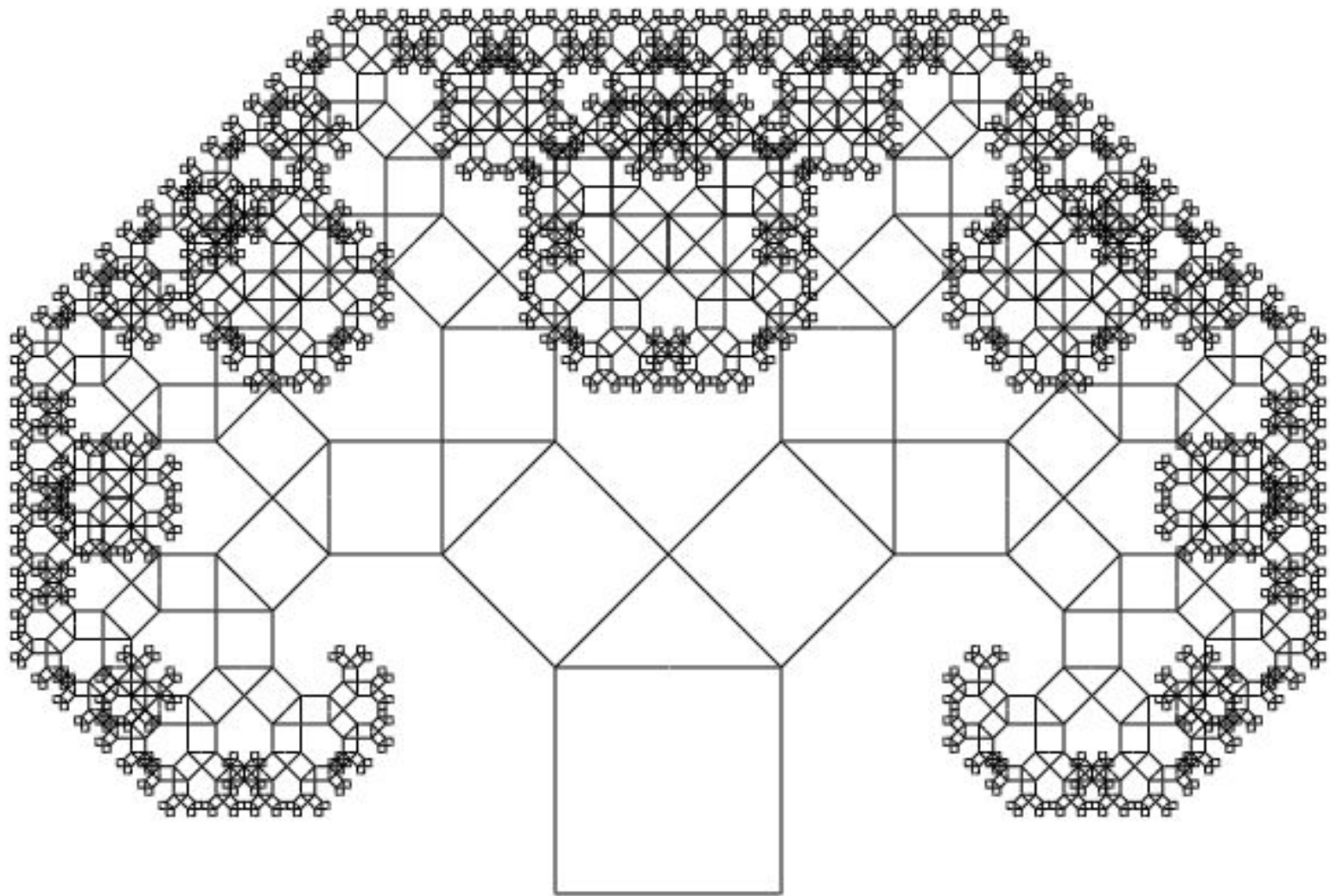
---

```

1 uses CRT, GraphABC;
2
3 Procedure Rect(x1, y1, l: Integer; a1: Real);
4 Begin
5     MoveTo(x1, y1);
6     LineTo(x1 + Round(l * cos(a1)), y1 - Round(l * sin(a1)));
7     LineTo(x1 + Round(l * sqrt(2) * cos(a1 + pi/4)),
8         y1 - Round(l * sqrt(2) * sin(a1 + pi/4)));
9     LineTo(x1 + Round(l * cos(a1 + pi/2)), y1 - Round(l * sin(a1 + pi/2)));
10    LineTo(x1, y1)
11 End;
12
13 Procedure Draw(x, y, l, a: Real);
14 Begin
15     If l > 4 Then
16     Begin
17         Rect(Round(x), Round(y), Round(l), a);
18         Draw(x - l*sin(a), y - l * cos(a), l / sqrt(2), a + pi / 4);
19         Draw(
20             x - l * sin(a) + l / sqrt(2) * cos(a + pi/4),
21             y - l * cos(a) - l / sqrt(2) * sin(a + pi/4),
22             l / sqrt(2),
23             a - pi/4)
24     End
25 End;
26 Begin
27     SetWindowCaption('Фракталы: Дерево Пифагора');
28     SetWindowSize(730,500);
29     ClearWindow;
30     Draw(280, 460, 100, 0);
31     Repeat Until KeyPressed
32 End.

```





# КОВЕР СЕРПИНСКОГО

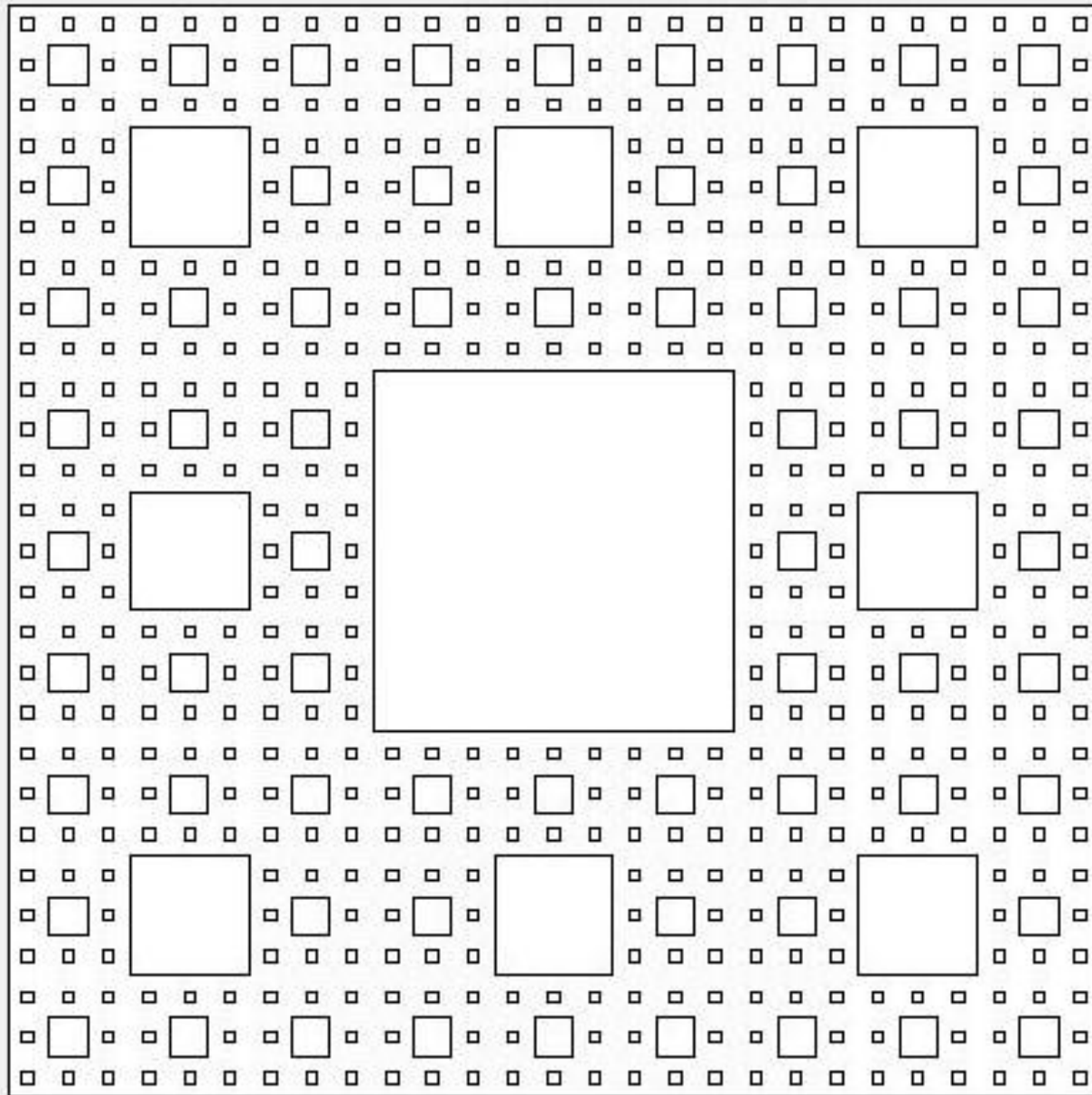
---



```

1 Uses CRT, GraphABC;
2 Const Z = 6; {Глубина фрактала}
3 Var
4   x1, y1, x2, y2, x3, y3: Real;
5
6 Procedure Serp(x1, y1, x2, y2: Real; n: Integer);
7 Var
8   x1n, y1n, x2n, y2n: Real;
9 Begin
10  If n > 0 Then
11    Begin
12      x1n := 2*x1/3 + x2 / 3;
13      x2n := x1/3 + 2*x2 / 3;
14      y1n := 2*y1/3 + y2 / 3;
15      y2n := y1/3+2*y2 / 3;
16      Rectangle(Round(x1n), Round(y1n), Round(x2n), Round(y2n));
17      Serp(x1, y1, x1n, y1n, n-1);
18      Serp(x1n, y1, x2n, y1n, n-1);
19      Serp(x2n, y1, x2, y1n, n-1);
20      Serp(x1, y1n, x1n, y2n, n-1);
21      Serp(x2n, y1n, x2, y2n, n-1);
22      Serp(x1, y2n, x1n, y2, n-1);
23      Serp(x1n, y2n, x2n, y2, n-1);
24      Serp(x2n, y2n, x2, y2, n-1)
25    End
26  End;
27 Begin
28   SetWindowCaption('Фракталы: Ковер Серпинского');
29   SetWindowSize(500,500);
30   ClearWindow;
31   Rectangle(20, 20, 460, 460);
32   Serp(20, 20, 460, 460, Z);
33   Repeat Until Keypressed
34 End.

```





# ТРЕУГОЛЬНИК СЕРПИНСКОГО

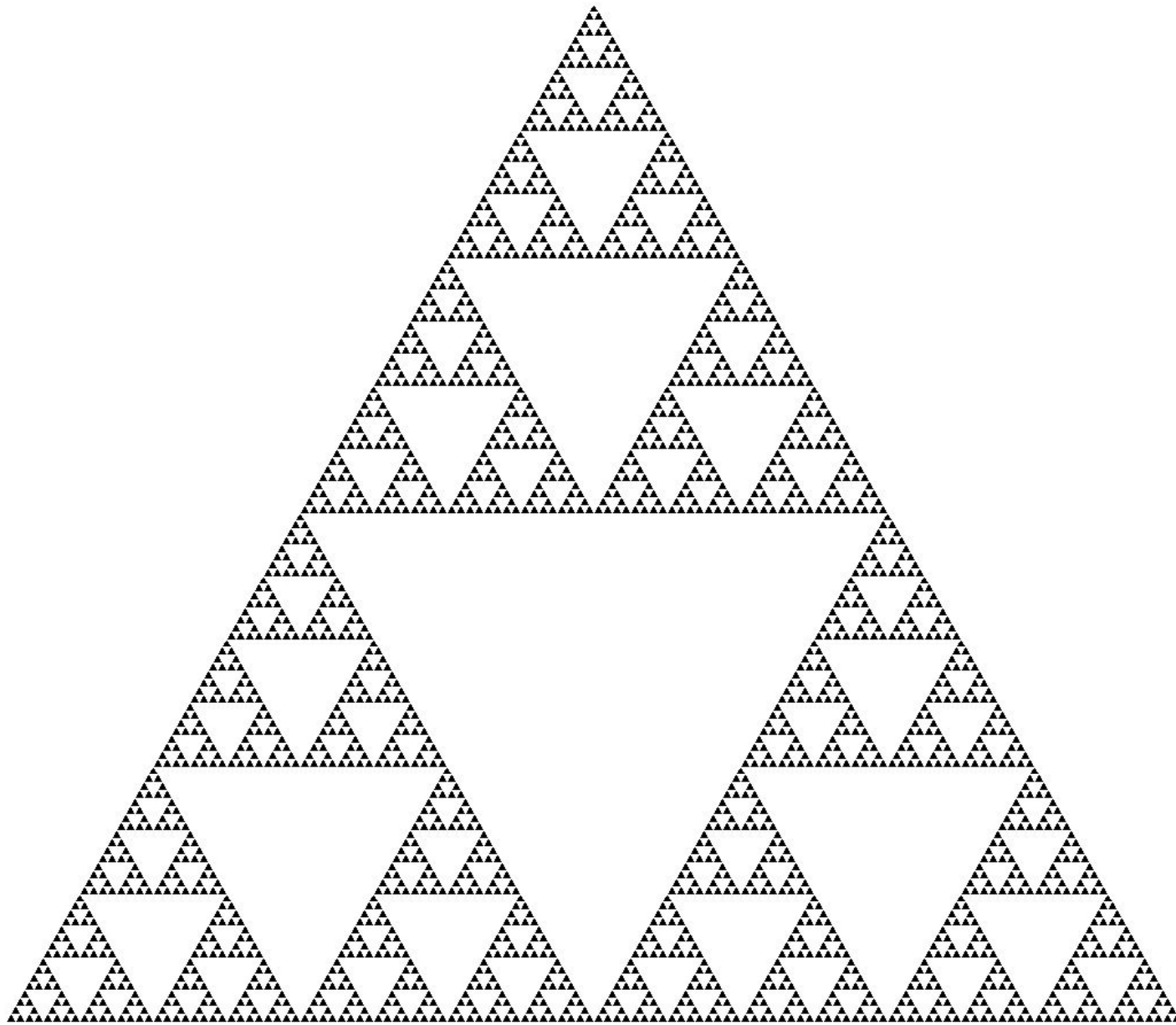
---

```

1 Uses CRT, GraphABC;
2 Const Z = 7; {Глубина фрактала}
3
4 Procedure tr(x1, y1, x2, y2, x3, y3: Real);
5 Begin
6     Line(Round(x1), Round(y1), Round(x2), Round(y2));
7     Line(Round(x2), Round(y2), Round(x3), Round(y3));
8     Line(Round(x3), Round(y3), Round(x1), Round(y1));
9 End;
10
11 Procedure draw(x1, y1, x2, y2, x3, y3: Real; n: Integer);
12 Var
13     x1n, y1n, x2n, y2n, x3n, y3n : Real;
14 Begin
15     If n > 0 Then
16     Begin
17         x1n := (x1 + x2) / 2;
18         y1n := (y1 + y2) / 2;
19         x2n := (x2 + x3) / 2;
20         y2n := (y2 + y3) / 2;
21         x3n := (x3 + x1) / 2;
22         y3n := (y3 + y1) / 2;
23         tr(x1n, y1n, x2n, y2n, x3n, y3n);
24         draw(x1, y1, x1n, y1n, x3n, y3n, n - 1);
25         draw(x2, y2, x1n, y1n, x2n, y2n, n - 1);
26         draw(x3, y3, x2n, y2n, x3n, y3n, n - 1);
27     End;
28 End;
29 Begin
30     SetWindowCaption('Фракталы: Треугольник Серпинского');
31     SetWindowSize(650, 600);
32     ClearWindow;
33     tr(320, 10, 600, 470, 40, 470);
34     draw(320, 10, 600, 470, 40, 470, Z);
35     Repeat Until KeyPressed;
36 End.

```





---

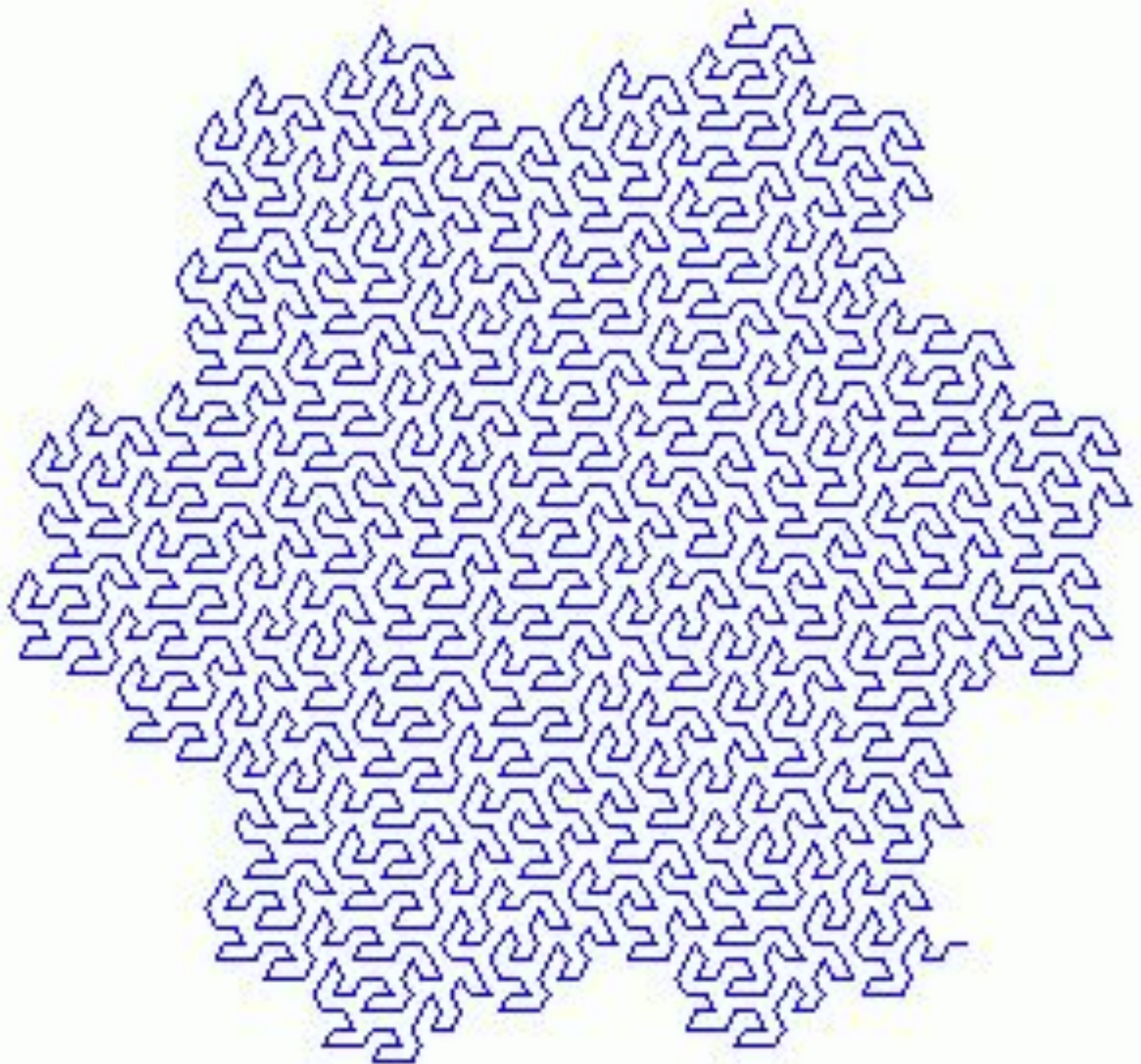
# **ЭКСПЕРИМЕНТАЛЬНОЕ МОДЕЛИРОВАНИЕ СОБСТВЕННЫХ ФРАКТАЛОВ**



```

1 Program GraphABC;
2
3 Uses CRT, GraphABC;
4
5 Procedure Draw(x, y, l, u : Real; t, p : Integer);
6
7 Procedure Draw2(x, y : Real; l, u : Real; t, p :
8 Integer);
9
10 Begin
11   Draw(x, y, l, u, t, p);
12   x := x + l*cos(u);
13   y := y - l*sin(u)
14 End;
15
16 Begin
17   If t > 0 Then t f
18     Begin
19       If p = 1 Then
20         Begin
21           x := x + l*cos(u);
22           y := y - l*sin(u);
23           u := u + pi
24         End;
25         u := u - 2*pi/19;
26         l := l*(7);
27         Draw2(x, y, l, u, t-1, 0);
28         Draw2(x, y, l, u+pi/3, t-1, 1);
29         Draw2(x, y, l, u+pi, t-1, 1);
30         Draw2(x, y, l, u+2*pi/3, t-1, 0);
31         Draw2(x, y, l, u, t-1, 0);
32         Draw2(x, y, l, u, t-1, 0);
33         Draw2(x, y, l, u-pi/3, t-1, 1)
34       End
35     Else
36       Line(Round(x), Round(y), Round(x + cos(u)*l),
37 Round(y - sin(u)*l))
38     End;
39   p
40 Begin
41   SetWindowCaption('График функции');
42   SetWindowSize(650, 500);
43   ClearWindow;
44   Draw(100, 355, 400, 0, 4, 0);
45   Repeat Until KeyPressed
46 End.

```





# ЗАКЛЮЧЕНИЕ

- изучены понятия, «фрактал», «Самоподобие, самоподобный объект», «язык программирования».
- выявлены существующие приемы и способы программирования фракталов;
- проанализированы программы для построения фракталов, с использованием изученных способов.