**Практическая работа №2**

**Фракталы и фрактальная графика**

1. Что такое фрактал? Какие виды бывают?

Фрактал – это фигура, обладающая свойством самоподобия. Объект называют самоподобным, если одна или более его частей похожа на его целое. При этом количество повторяющихся частей у фрактала стремится к бесконечности — этим он отличается от самоподобных геометрических фигур с конечным числом звеньев (предфракталов).

Виды:

**Геометрические**— строятся на основе исходной фигуры, которая определённым образом делится и преобразуется на каждой итерации.

**Алгебраические**— строятся на основе алгебраических формул.

**Стохастические** — образуются в том случае, если в итерационной системе случайным образом изменяется один или несколько параметров.

1. Что такое ломанная-генератор?

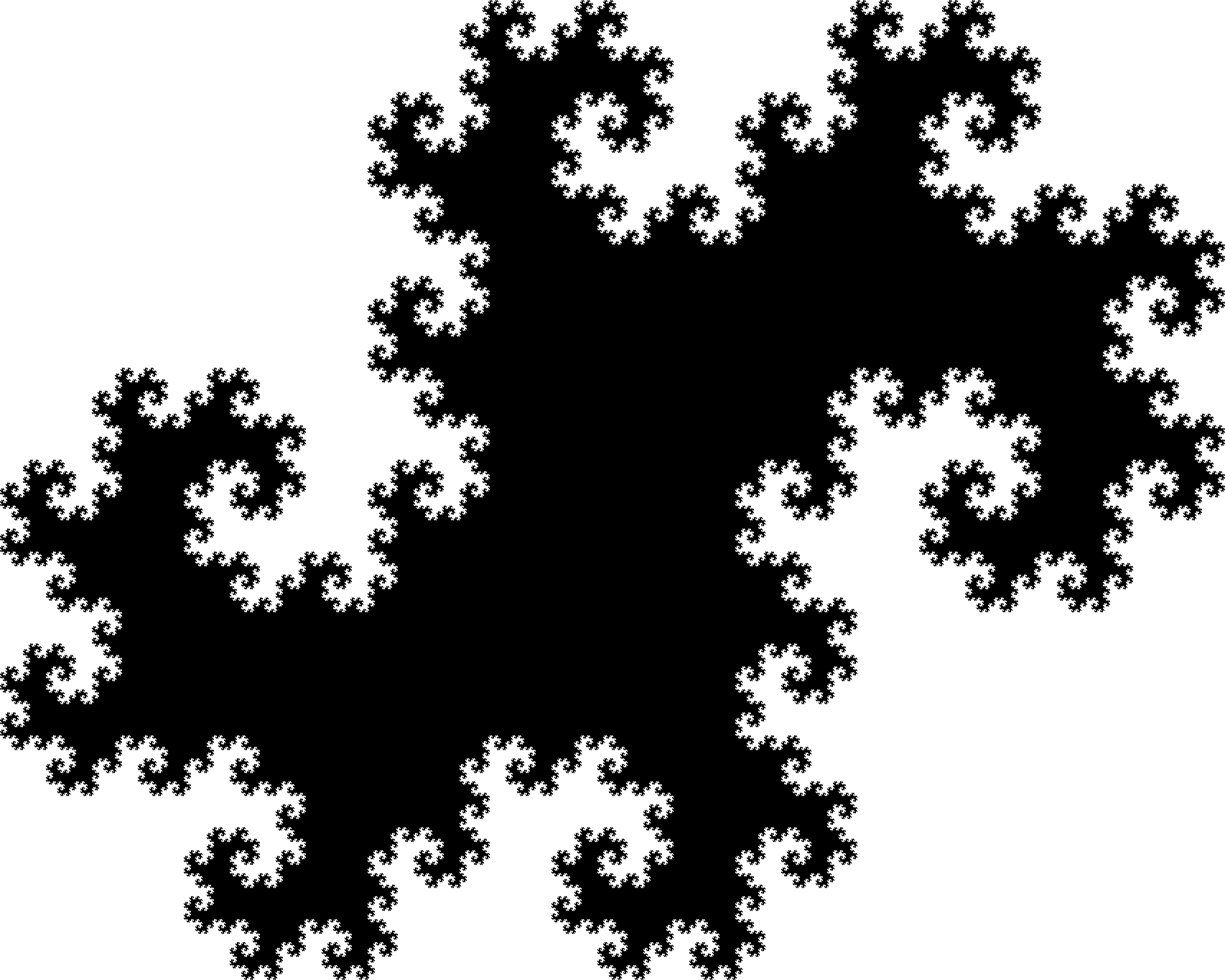
Ломанная-генератор представляет собой алгоритм, создающий ломаную или кривую на плоскости, состоящую из участков прямых линий, соединенных в узлах. За один шаг алгоритма каждый из отрезков, составляющих ломаную, заменяется на ломаную-генератор в соответствующем масштабе. В результате бесконечного повторения этой процедуры получается геометрический фрактал.

1. Опишите фрактал из своего варианта ДКР.

Фрактал Дракон Хартера-Хейтуэя (Кривая Дракона)

Кривая была придумана физиком Джон Хейтуэй, а теорию разработали Вильям Хартер, Хейтуэй и Брюс Бенкс.

Фрактальная кривая Дракона является одной из известных фрактальных кривых, получившей свое название из-за своего характерного "чешуйчатого" вида, напоминающего расположение чешуи на теле дракона. Кривая Дракона обладает фрактальной размерностью, что означает, что она заполняет плоскость, но при этом имеет бесконечную длину.



1. Укажите ломанную-генератор и опишите алгоритм построения фрактала в зависимости от глубины.

Алгоритм построения фрактала кривой Дракона варьируется в зависимости от глубины следующим образом:

1. При глубине 0 начинаем с одного сегмента.

2. При глубине 1 создаем базовую структуру кривой Дракона, применяя "развилку и поворот" к начальному сегменту, получая два сегмента, один из которых повернут на 90 градусов.

3. При глубине 2 повторяем процесс для каждого из двух сегментов, полученных на предыдущем шаге, заменяя каждый сегмент на два новых сегмента, один из которых повернут на 90 градусов, и так далее.

4. Этот процесс продолжается с увеличением глубины, при этом количество сегментов и их ориентация меняются, определяя структуру кривой Дракона на каждом уровне глубины.



1. Запишите рекурсивную подпрограмму для построения фрактала, в ней укажите рекурсивную триаду.

**procedure** DrawDragonCurve(x, y, length, angle, depth: integer);

**begin**

**if** depth = 0 **then**

**begin**

moveto(x, y);

angle := angle **mod** 4;

**case** angle **of**

0: lineto(x + length, y);

1: lineto(x, y + length);

2: lineto(x - length, y);

3: lineto(x, y - length);

**end**;

**end**

**else**

**begin**

length := round(length / sqrt(2));

DrawDragonCurve(x, y, length, angle, depth - 1);

angle := (angle + 1) **mod** 4;

DrawDragonCurve(x + round(length \* cos(pi / 4 \* angle)), y + round(length \* sin(pi / 4 \* angle)), length, angle, depth - 1);

**end**;

Рекурсивная триада:

1. Базовый случай

**if** depth = 0 **then**

**begin**

moveto(x, y);

angle := angle **mod** 4;

**case** angle **of**

0: lineto(x + length, y);

1: lineto(x, y + length);

2: lineto(x - length, y);

3: lineto(x, y - length);

**end**;

1. Параметризация:

- x, y: координаты начальной точки рисования кривой

- length: длина отрезка при каждой итерации

- angle: угол, определяющий направление отрезка (возможные значения 0, 1, 2, 3)

- depth: глубина рекурсии, определяющая степень детализации кривой

1. Декомпозиция

Рекурсивный случай включает в себя следующие шаги:

1. Уменьшение длины отрезка: length := round(length / sqrt(2));

2. Вызов рекурсивной подпрограммы для первой половины кривой:

DrawDragonCurve(x, y, length, angle, depth - 1);

3. Изменение угла: angle := (angle + 1) mod 4;

4. Вызов рекурсивной подпрограммы для второй половины кривой:

DrawDragonCurve(x + round(length \* cos(pi / 4 \* angle)), y + round(length \* sin(pi / 4 \* angle)), length, angle, depth - 1);