Фракталы

Это фигура, у которой Хауздорфова размерность больше ее Топологической размерности.

***Хауздорфова размерность*** – размерность, согласующаяся с общими представлениями о размерности. Гладкая кривая имеет размерность 1, треугольник 2, объемная область – 3. Для фрактальных множеств размерность может быть дробной.

***Топологическая размерность*** или ***Размерность Лебега*** – ?

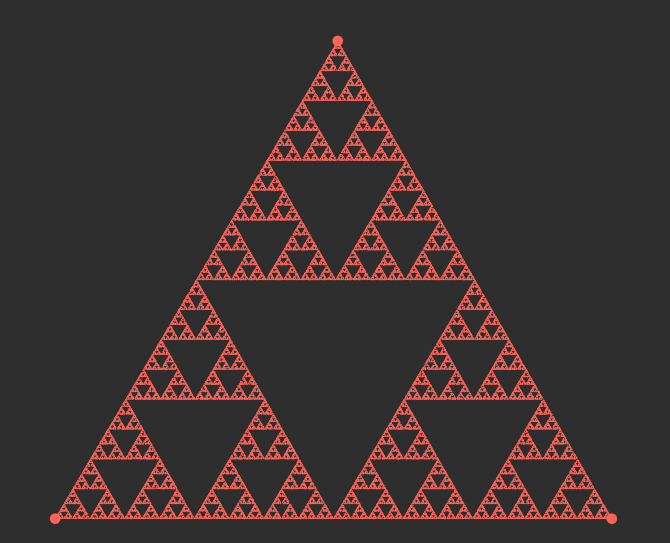
***Мера Лебега*** – это мера пространства данной n-мерной фигуры (для отрезка – это длина, для многоугольника или круга – это площадь), n ∈ **N**

◉ Все фракталы с размерностью Хауздорфа от 1 до 2 имеют бесконечную длину и нулевую площадь

Фракталы используются для создания эффективной формы антенн.

# Известные фракталы

## ▣ Треугольник серпинского:

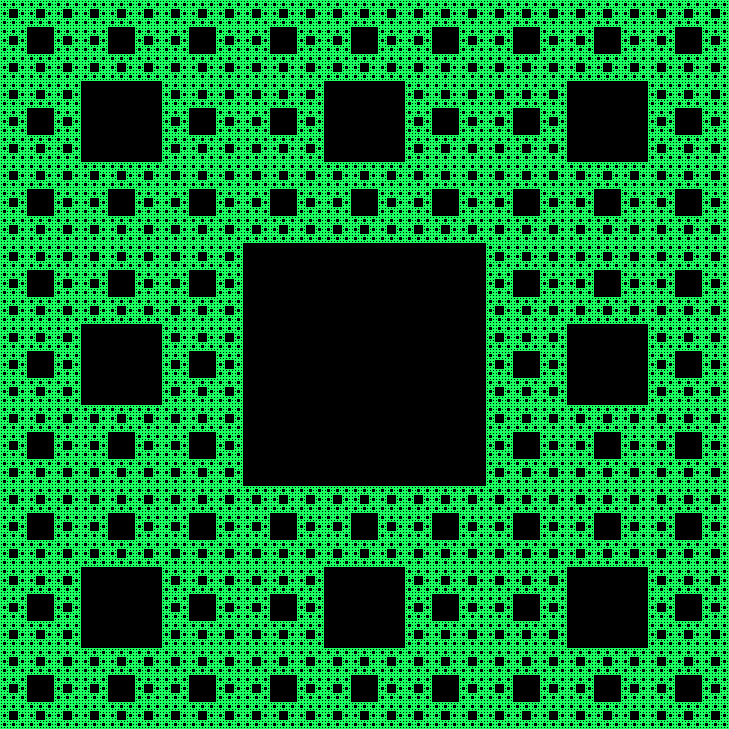


Хауздорфова размерность = log2 3 ≈ 1.585

Топологическая размерность = 1

Мера Лебега = 0

## ▣ Ковер (квадрат) Серпинского

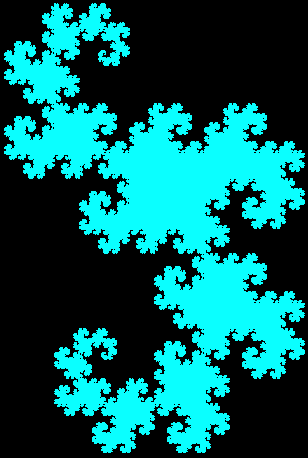


Хауздорфова размерность = log3 8 ≈ 1.89

Топологическая размерность = 1

Мера Лебега = 0

## ▣ Кривая дракона (Дракон Хартера-Хейтуэя)



Хауздорфова размерность = 2

Способы получения:

Фрактал может быть записан как L-система с параметрами:

угол равен 90° или pi/2

начальная строка — FX

правила преобразования строк:

X → X+YF+

Y → -FX-Y

Берём отрезок, сгибаем его пополам. Затем многократно повторяем итерацию. Если после этого снова разогнуть получившуюся (сложенную) линию так, чтобы все углы были равны 90°, мы получим драконову ломаную.

**void** dragon\_func(**int** x1, **int** y1, **int** x2, **int** y2, **int** n)

{

**int** xn,yn;

Graphics g = Graphics.FromHwnd(pictureBox1.Handle);

var drawingPen = new Pen(Brushes.Navy, 1);

**if**(n > 0)

{

xn = (x1 + x2) / 2 + (y2 - y1) / 2;

yn = (y1 + y2) / 2 - (x2 - x1) / 2 ;

dragon\_func(x2, y2, xn, yn, n - 1);

dragon\_func(x1, y1, xn, yn, n - 1);

}

var point1 = new Point(x1, y1);

var point2 = new Point(x2, y2);

g.DrawLine(drawingPen, point1, point2);

}

**private** **void** Draw\_dragon\_curve(**object** sender, EventArgs e)

{

**int** x1,y1,x2,y2, k;

x1 = 200;

y1 = 200;

x2 = 390;

y2 = 400;

k = 15;

dragon\_func(x1,y1,x2, y2, k);

}

**private** **void** Clear\_curve(**object** sender, EventArgs e)

{

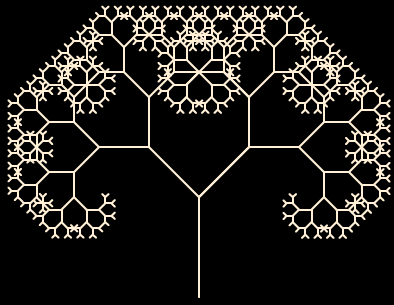
Graphics g = Graphics.FromHwnd(pictureBox1.Handle);

var bgcolor = new SolidBrush(Color.White);

g.FillRectangle(bgcolor, 0, 0, 660, 516);

}

## ▣ Дерево Пифагора



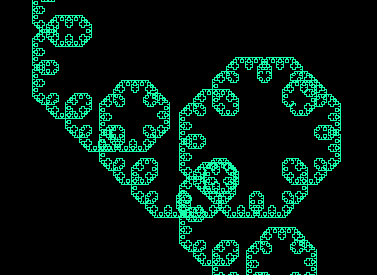
Размерность = 2 (если строить из прямоугольников)

Есть начальная точка из которой исходит первый отрезок – ***ствол***.

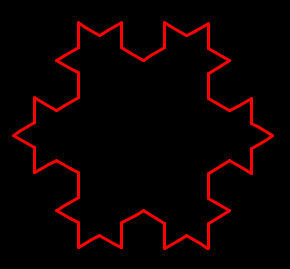
Параметры фрактала:

* количество стволов
* угол между двумя новыми ветвями
* количество подстволов (≥ 1)
* симметричность / ассиметричность ветвей (для каждого подствола свой угол)
* число, в которое длина предыдущей ветви больше длины следующей
* фигура ветви (линия, квадрат, перевернутый треугольник), при этом двумерные фигуры касаются друг друга углами. Можно также отрисовывать только контур двумерной фигуры.

## ▣ Кривая Леви



## ▣ Снежинка (кривая) Коха



Хауздорфова размерность = log3 4 ≈ 1.26

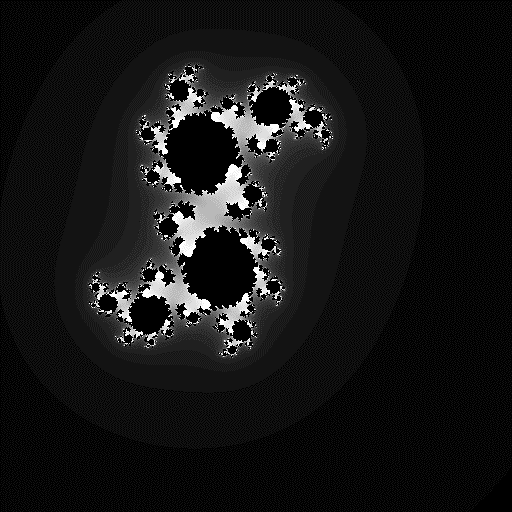
Множество Кантора (скучный фрактал)



Хауздорфова размерность = log3 2 ≈ 0.63

Ну ваще можно поиграть с коэффициентами и м б будет ок.

## ▣ Множества Жюлиа



Размерность =

Множество строится на основе функции:

f(z) = zp + c, z ∈ **C**

и спомощью итерационной формулы:

zn+1 = (zn)p + c

z0 = k, k ∈ **C**

Возведение двумерной точки в квадрат:

(x + yi)2 = x2 – y2 + 2xyi

(x, y) → (x2 – y2; 2̇xy)

Суть состоит в том, чтобы понять уходит ли данная (всякая около единичной окружности) точка в бесконечность или нет. Фактически нужно возвести каждую точку двумерной ограниченной плоскости в квадрат (как комплекстное число)

Степень p – это количество «вершин» множества.

При p → ∞ множество Жюлиа стремиться к кругу.

При возведении гиперкомплексного числа в степень, которое находится внутри n-мерной сферы, соответствующей мерности гиперкомплексного числа, это число будет стремиться к нулю. В противном случае модуль этого числа будет увеличиваться. Кроме того существует и 3-й случай, когда число находится на границе n-мерной сферы, тогда модуль этого числа будет постоянным.

Чтобы визуализировать множество, нужно окрашивать точки отдаляющиеся от нуля и точки приближающиеся к нулю по-разному. Чем стремительнее изменение точки с каждой итерацией, тем насыщенней или тусклее стоит окрашивать эту точку.

◉ Если константа c = 0, то множество Жюлиа представляет собой круг.

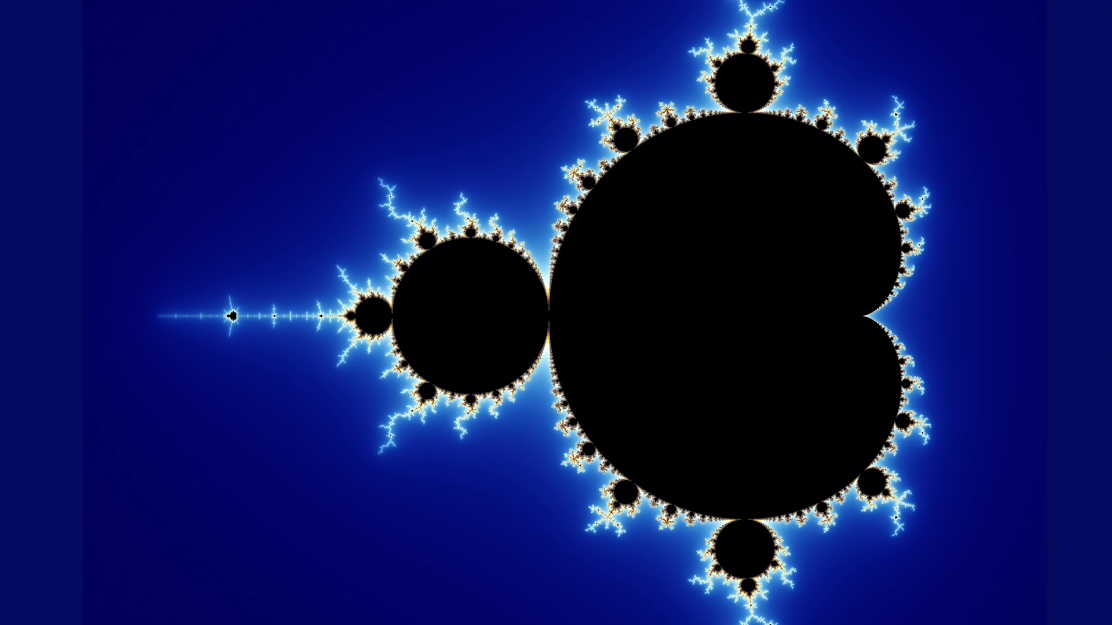
можно вычислять итерации формулы в обратном порядке:

zn-1 =

Множество Жюлиа не ограничивается формулой с простым полиномом – можно брать синус, косинус, экспоненту, умножать на константу и т. д.

<https://acko.net/blog/how-to-fold-a-julia-fractal/>

## ▣ Множество Мандельброта



Размерность = 2

Данный фрактал представляет собой множество всех возможных связных множеств Жюлиа. В формуле z1 = 0, а константа ***c*** теперь параметр. Множество Мандельброта проходит по всем значениям параметра ***c***, поэтому оно содержит все множества Жюлиа для которых определенное значение параметра ***c*** было константой.

Формула для множества Мандельброта:

zn+1 = zn2 + zn – ? Onigiri

zn+1 = zn2 + c

Псевдокод:

For each point kon the complex plane do:

let x=0.

repeat infinite times:

x=x^2+k.

end repeat

if x goes to infinity,

k is not in the set. Color is white.

else

k is in the set. Color is black.

Если в любой конкретный момент вычислений, расстояние от Zn до начала координат больше 2, то мы можем принять, что данная точка(k) уйдет в бесконечность. (if x^2 > 2)

For each point k in the complex plane do:

let x=0.

repeat 4000 times

let x=x^2+k

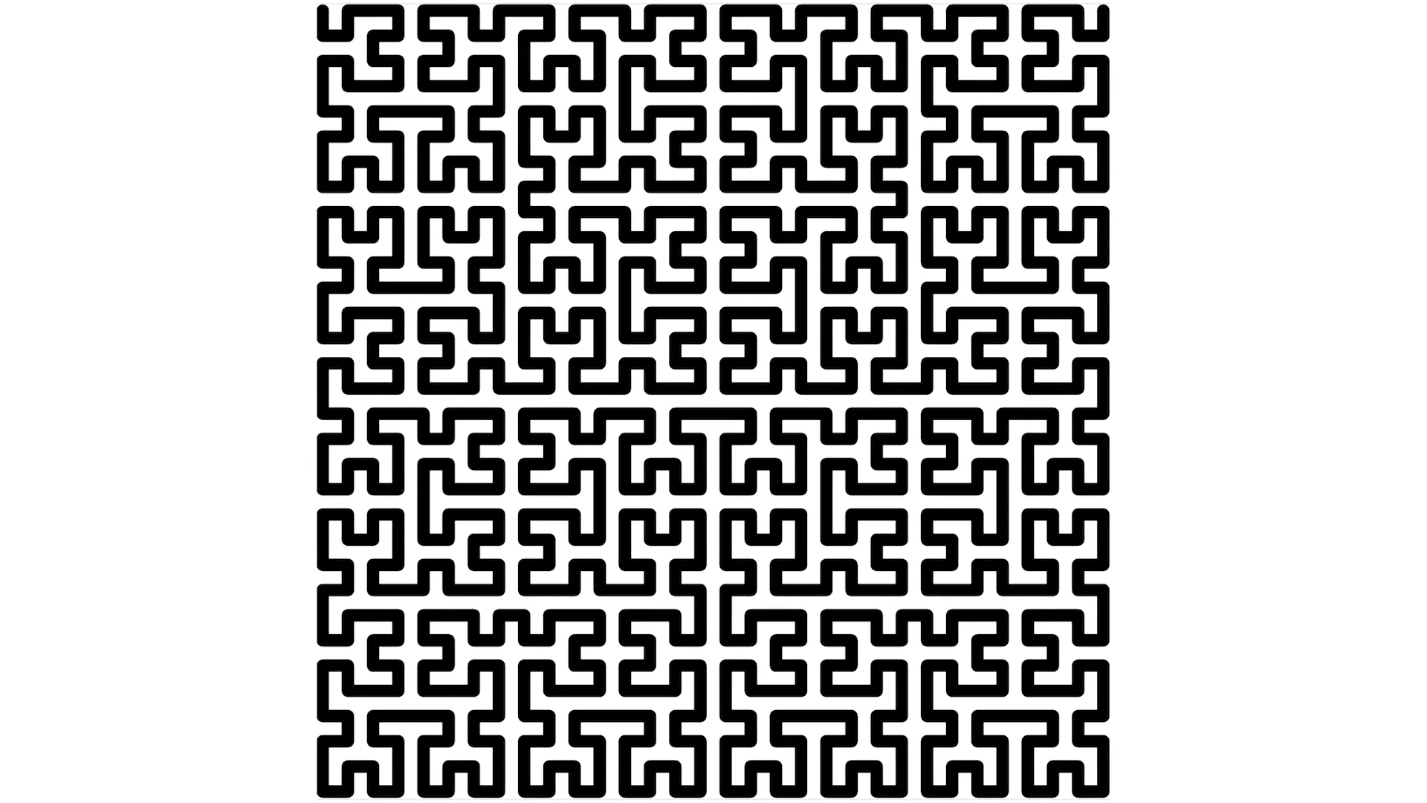
if x^2 > 4 then Color it white

Break.

end repeat

if we reached 4000 then Color it black.

## ▣ Кривая Гильберта



Размерность = 2

▣ Папоротники



Строятся на основе 4 простых преобразований с матрицами. Каждое преобразование выполняет свою функцию: рост стебля, образование левых и правых листьев.