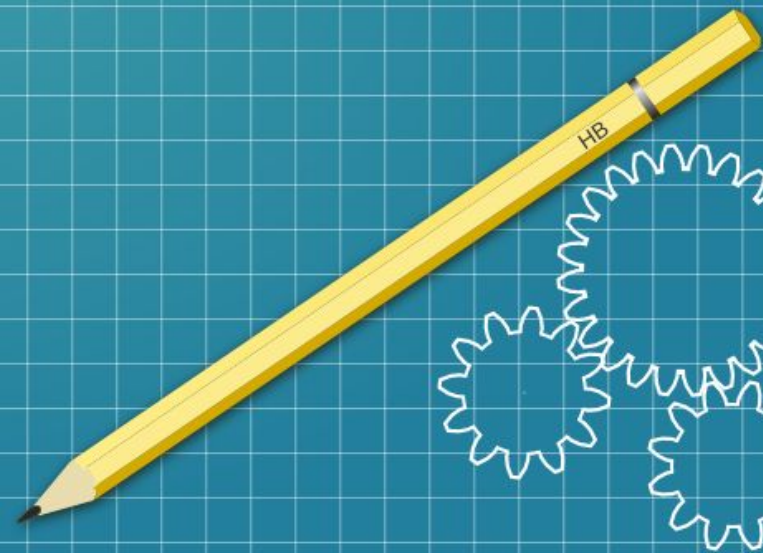
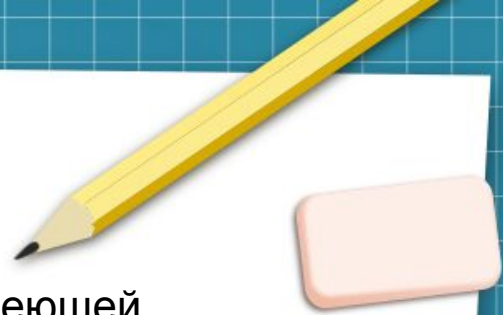


# Кривая Коха. Дробная размерность. Метод L-систем



# Кривая Коха

- 1904г. Хельге Фон Кох, статья «Об одной непрерывной кривой, не имеющей касательной, построенной с помощью методов элементарной геометрии»
- Кривая Коха нигде не дифференцируема и не спрямляема.
- Кривая Коха имеет бесконечную длину.
- Кривая Коха не имеет самопересечений.
- Кривая Коха имеет промежуточную хаусдорфову размерность, которая равна 1,26
- Длина кривой Коха описывается выражением
- $K_n = (4/3)^n \rightarrow \text{бесконечность}$



# СНЕЖИНКА КОХА



1



2



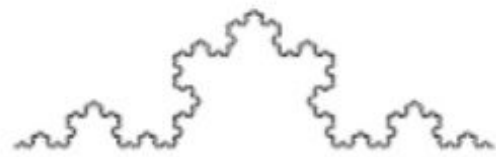
3



4



5



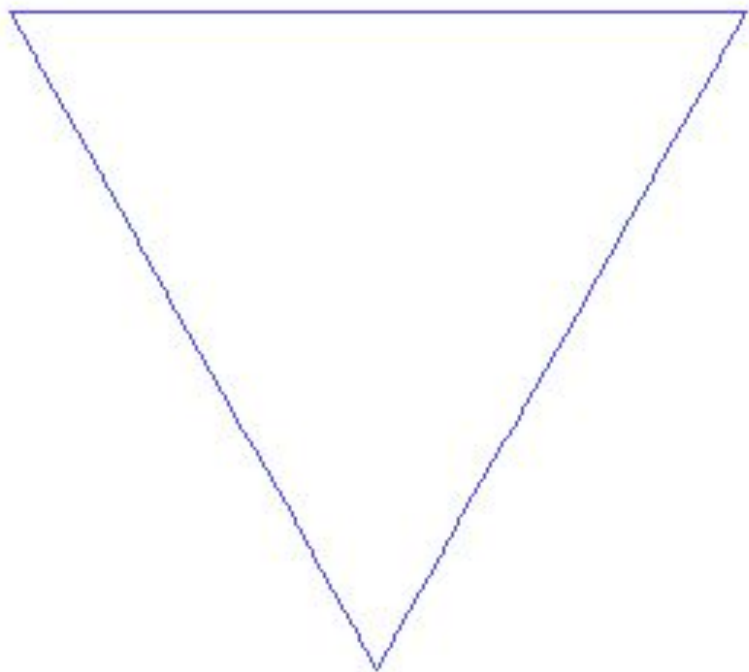
7

# Кривая Коха

```
• void DrawKoch( double dir, double len, int n)
• { double dirRad=0.0174533*dir;
•   if (n==0) lineRel(len*cos(dirRad), len*sin(dirRad));
•   else
•     { n--;
•       len/=3;
•       DrawKoch(dir,len,n);
•       dir+=60;
•       DrawKoch(dir,len,n);
•       dir-=120;
•       DrawKoch(dir, len,n);
•       dir+=60
•       DrawKoch(dir,len,n); }
• }
```

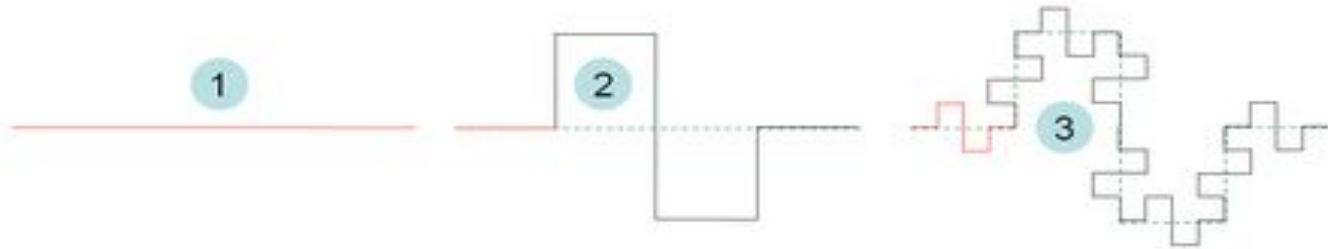


# Кривая Коха



# Вариации кривой Коха

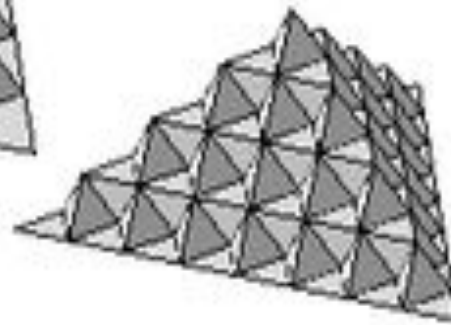
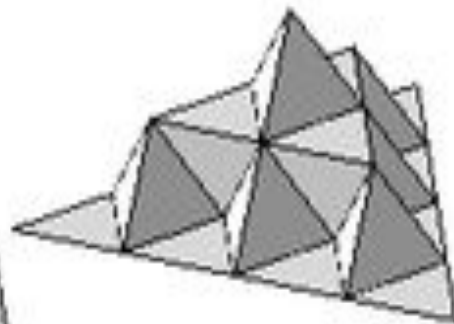
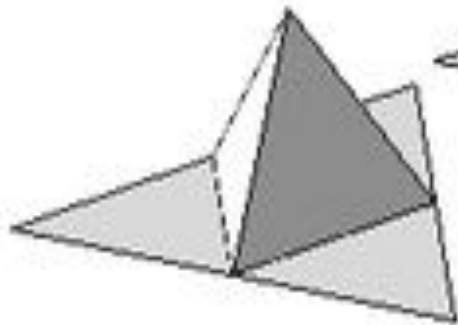
- Квадратичная кривая Коха



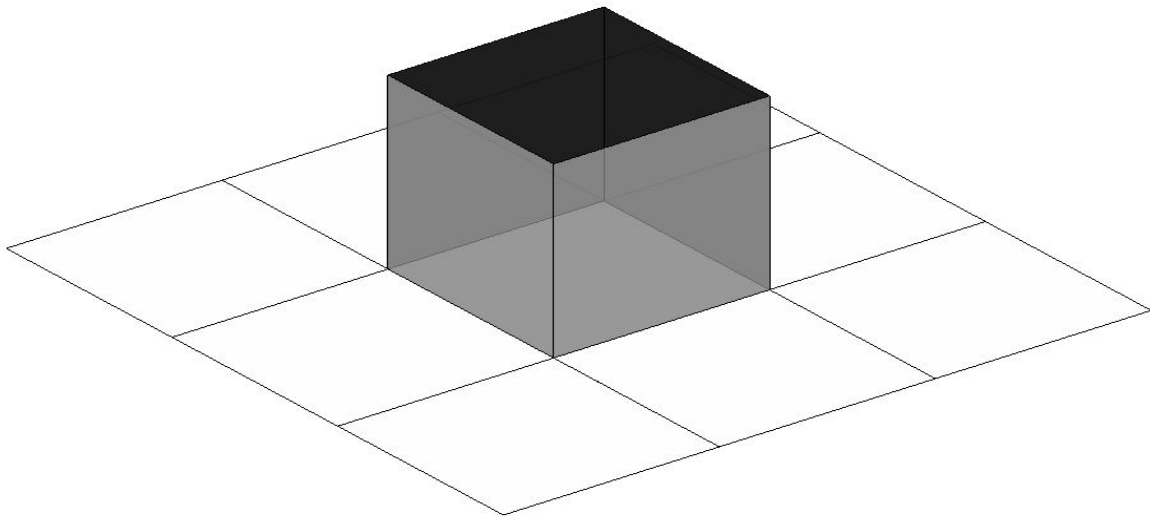


# Вариации кривой Коха

- Поверхность Коха

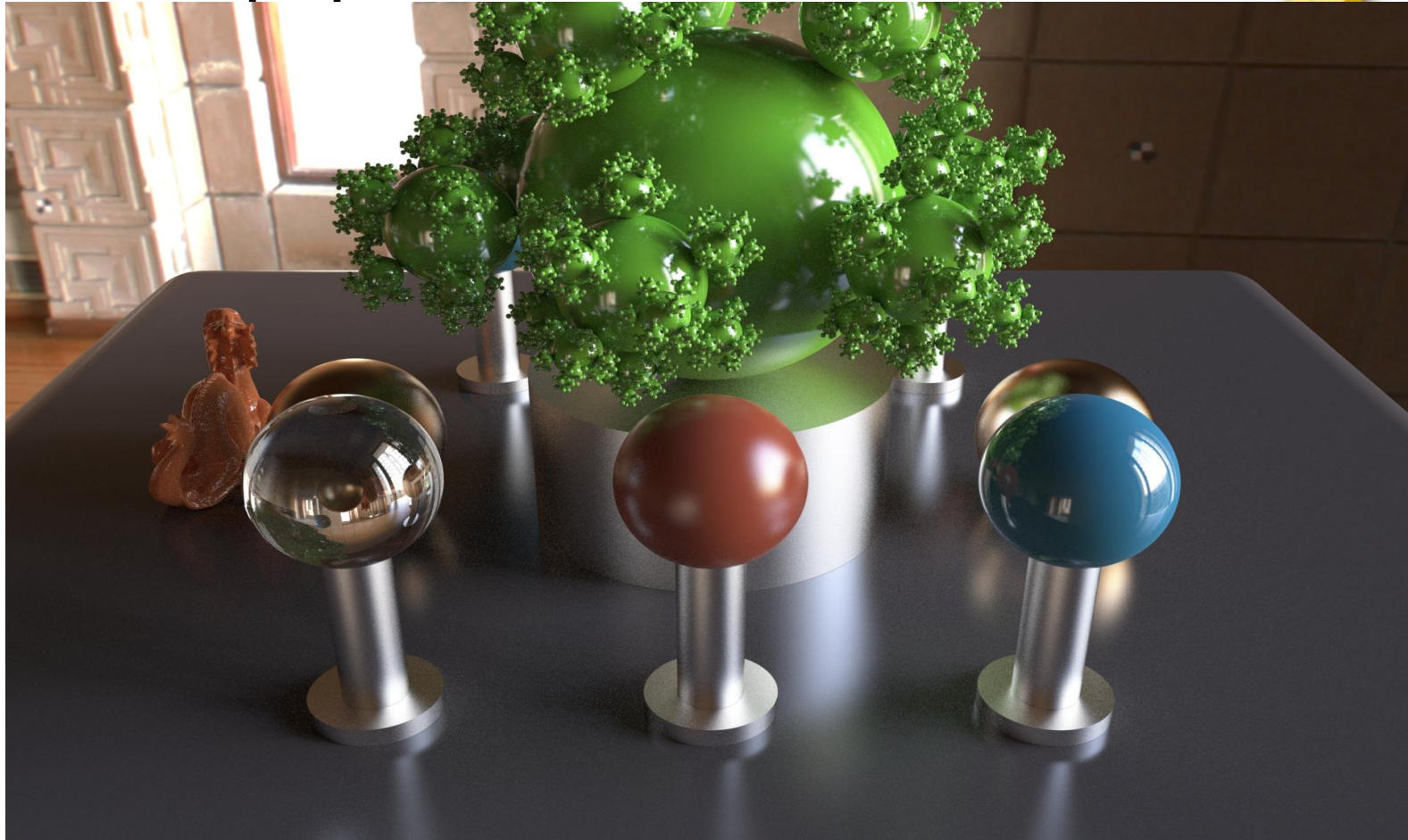


# Квадратичная поверхность Коха



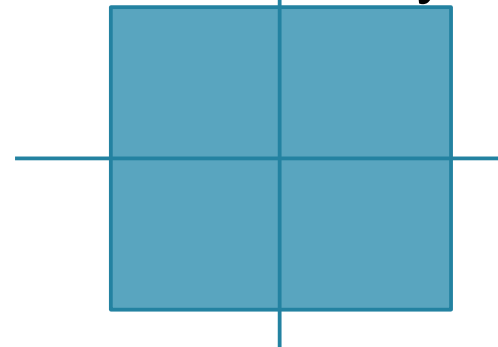
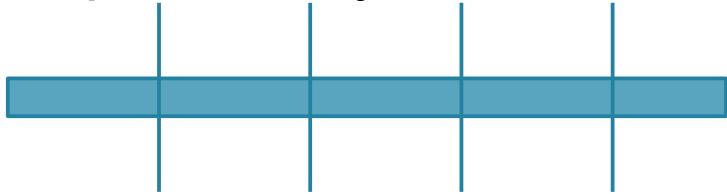


# Сферическая снежинка Хейнса



# Дробная размерность

- Когда речь идет об обычных геометрических объектах: линия, поверхность, шар, то их топологические размерности известны и являются целыми числами.
- Простой способ измерить длину кривых, площадь поверхности или объем тела состоит в том, чтобы разделить их на небольшие элементы – отрезки длиной  $1/n$ , квадраты со стороной  $1/\sqrt{n}$  или на небольшие кубы с ребрами  $1/\sqrt[3]{n}$

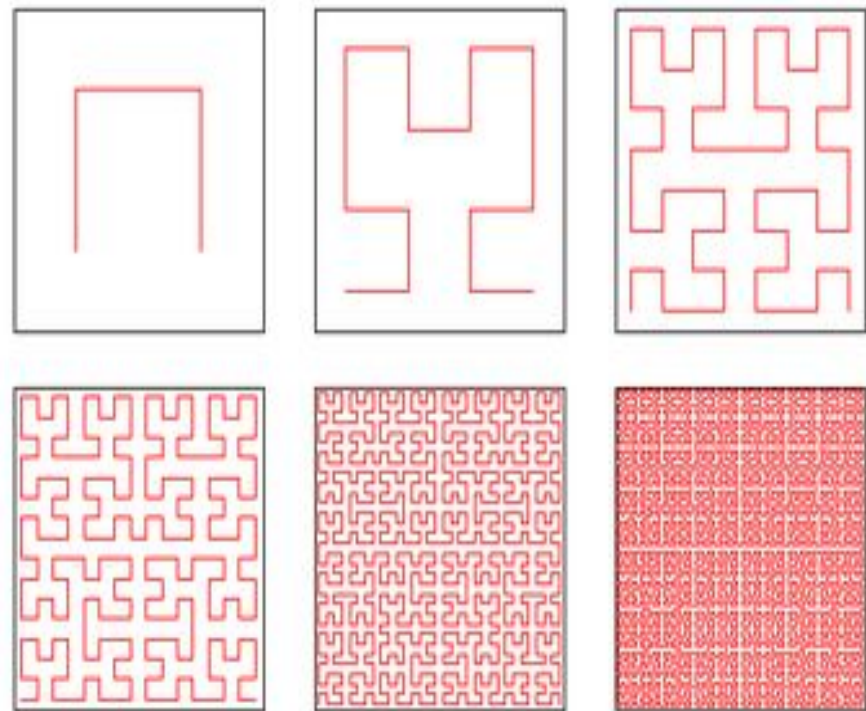


## Дробная размерность (Хаусдорф, 1919)

- Будем говорить, что объект имеет размерность  $D$ , если при делении его на  $N$  равных частей, каждая часть будет иметь сторону меньшую чем сторона исходного объекта в  $r$  раз
- $r = (1/N)^{(1/D)}$
- Найдем отсюда  $D$
- $D = \log N / \log (1/r)$
- Здесь  $N$  – количество частей,  $r$  – отношение длины ребра маленького объекта к большому.

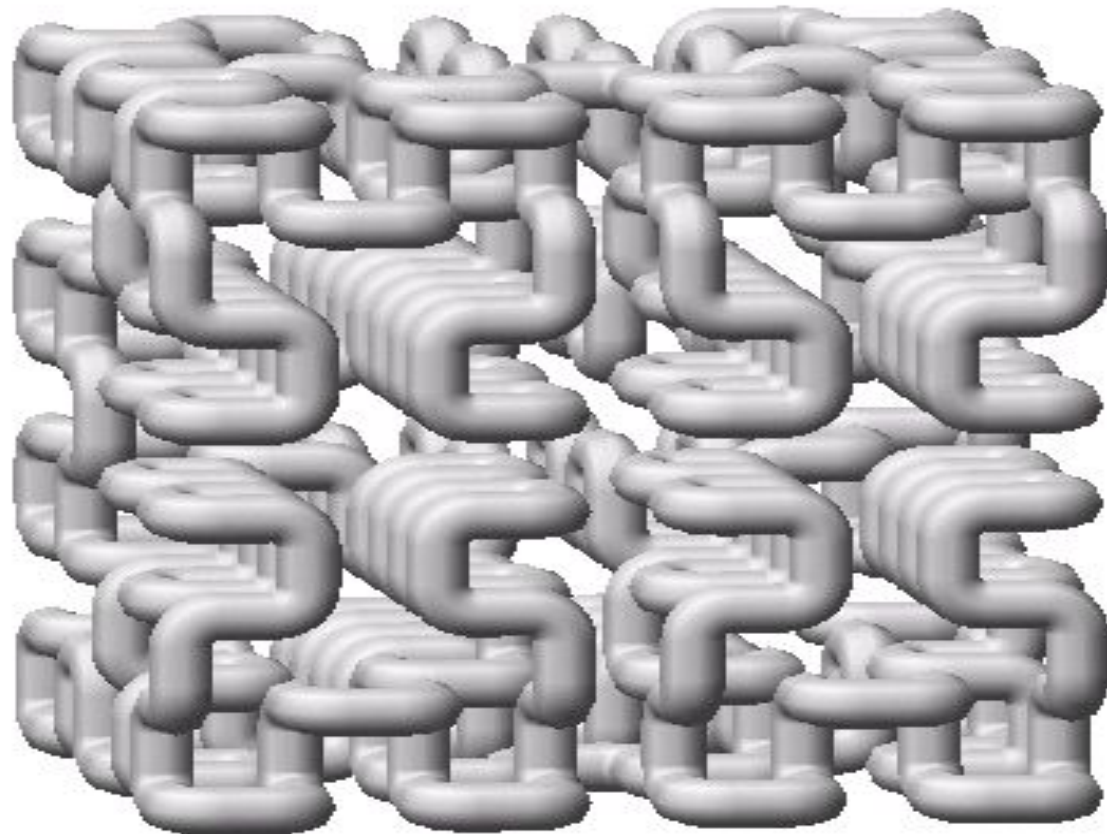
# Дробная размерность

- Для кривой Коха  $N=4$ ,  $r=1/3$
- $D=\ln 4/\ln 3= 1.26\dots$
- Кривая Гильберта (1891г)
- $D=\ln 4 / \ln 2 = 2$





# Трёхмерная кривая Гильберта



# Кривая Гильберта



- Используется для выявления ошибок при передаче данных
- Числа от 0 до 7 кодируются 000, 001, 010, 011, 100, 101, 110, 111
- Каждое из чисел можно расположить в вершине единичного куба. Например, 001 –  $(0,0,1)$
- Если упорядочить числа, следуя кривой Гильберта, получим код Грея
- Код Грея применяется для кодирования информации в сетях цифрового телевидения



# Кривая Гильберта



- Используется для цифровой обработки изображений
- Например, для распечатки изображения в градациях серого при ограниченной палитре оттенков.
- При проходе палитры по кривой Гильберта отсутствуют дефекты изображения.

# Метод L-систем

- „F“ – forward(1,1)
- „+“ – turn(A)
- „-“ – turn(-A)
- Для  $A=60$
- $S_1 = \text{„F-F++F-F“}$
- $F \rightarrow \text{„F-F++F-F“}$
- $S_2 = \text{“(F-F++F-F)-(F-F++F-F)++(F-F++F-F)-(F-F++F-F)”}$



# Метод L-систем

- FILE \*f1,\*f2;
- for(; ch!=EOF)
- {
- ch=fgetc(f1);
- if(ch=="+"|| ch=="-") fputc(ch,f2);
- else if(ch=="F") fputs(f2, „F-F++F-F“);
- }



# Метод L-систем

- Подпрограмма для черепахи
- for( each ch from f2)
- {
- If (ch=="+") turn(A);
- else if (ch=="-") turn(-A);
- else if (ch=="F") forward(1,1);
- }



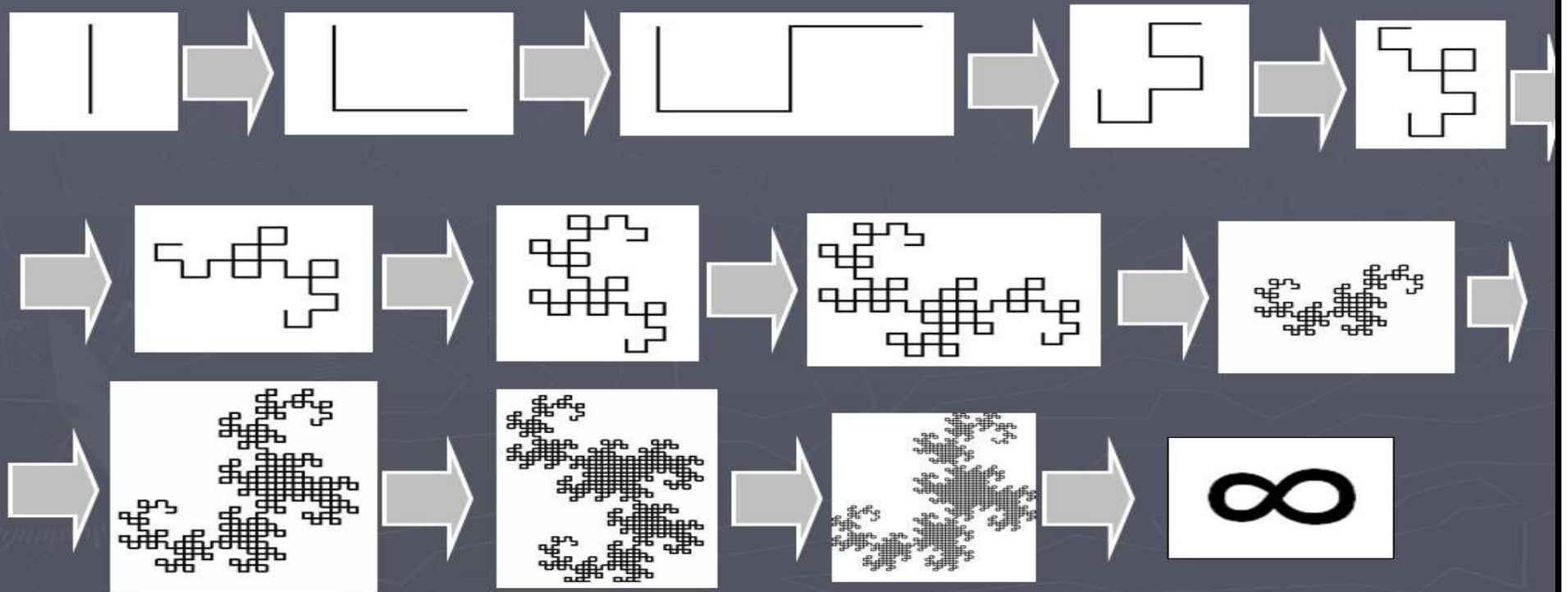
# Метод L-систем

- „F“  $\rightarrow$  „F“
- „X“  $\rightarrow$  „X+YF+“
- „Y“  $\rightarrow$  „-FX-Y“
- Если начальная строка atom=FX, то
- $s_1 = FX + YF +$
- $s_2 = F(X + YF +) + (-FX - Y)F +$
- Что будет рисовать черепаха?
- $s_1 = F + F +$ ,  $s_2 = F + F + + -F - F +$



# Метод L-систем

## “Кривая дракона” Э. Хейуэея





!

0



# Метод L-систем

- void produceString( char \*str, int order)
- { for(; \*str; str++)
- { switch (\*str)
- { case ' +': CD=A; break'
- case „-“: CD=-A; break;
- case „F“: if (order>0) produceString(Fstr, order-1);
- else forward(len, 1); break;
- case „X“ : if(order>0) produceString(Xstr, order-1);
- case „Y“: if(order>0) produceString(Ystr, order-1);}
- }
- }



# Метод L-систем

- „[“ – затолкнуть в стек
- „]“ – вытолкнуть из стека
- // добавить в функцию produceString()
- case „[“: saveTurtle(); break;
- case „]“: restoreTurtle(); break;
- Куст
- $F \rightarrow "FF-[-F+F+F]+[+F-F-F]"$

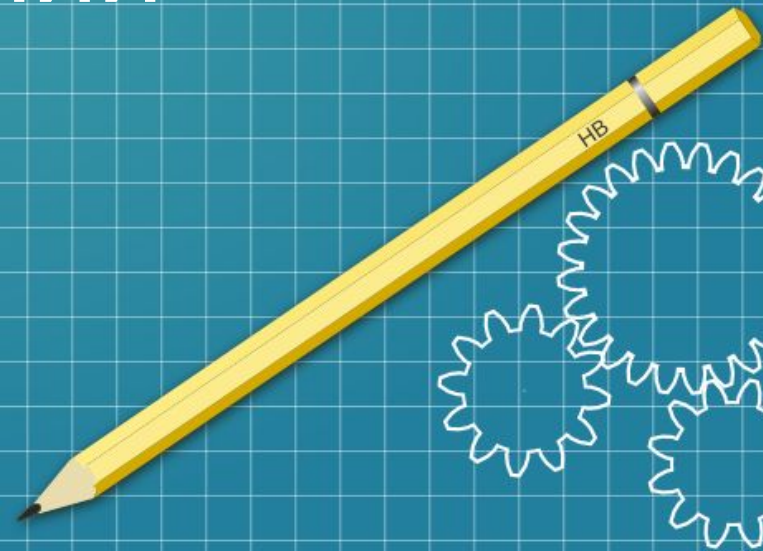


# Задание



- Записать инструкции для черепахи в виде L-строк для одной из квадратичных кривых Коха
- Вычислить размерность одной из квадратичных кривых Коха или дракона
- Записать и нарисовать первые три поколения кривой
- $atom=YF$ ,  $Fstr= F$ ,  $Xstr= YF+XF+Y$ ,  $Ystr=XF-YX-X$ , угол  $A=60$
- Нарисовать фрактал куст( три итерации, угол  $A=22.5$ )
- $F \rightarrow FF-[-F+F+F]+[+F-F-F]$

# Рептилии



# Рептилии

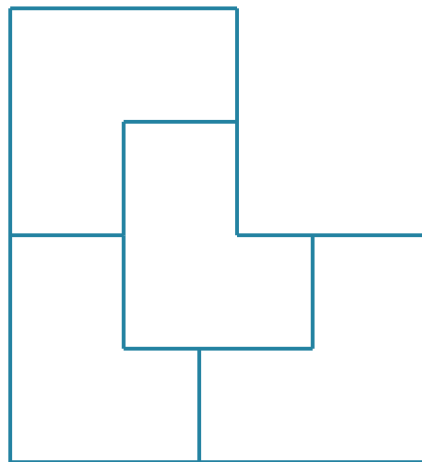
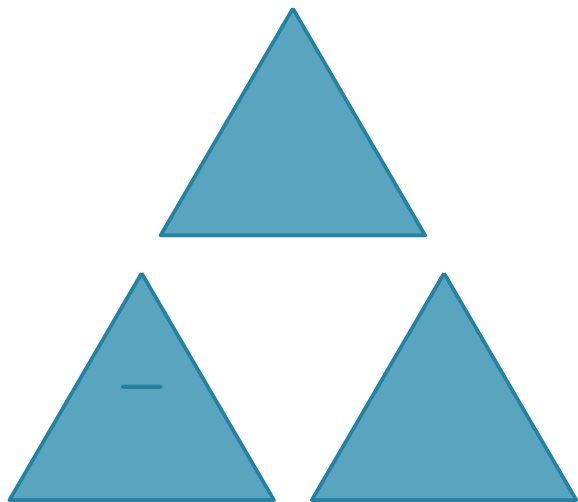
- Класс неперiodических мозаик
- Рисуются от большого к малому или наоборот
- Различные копии рептилии совмещаются друг с другом, образуя большую рептилию
- `void trio( double size, int depth)`
- `{ if (depth==1) draw();`
- `else for( int i=0; i<4; i++)`
- `{ draw1( size/2, depth-1);`
- `}`
- `}`





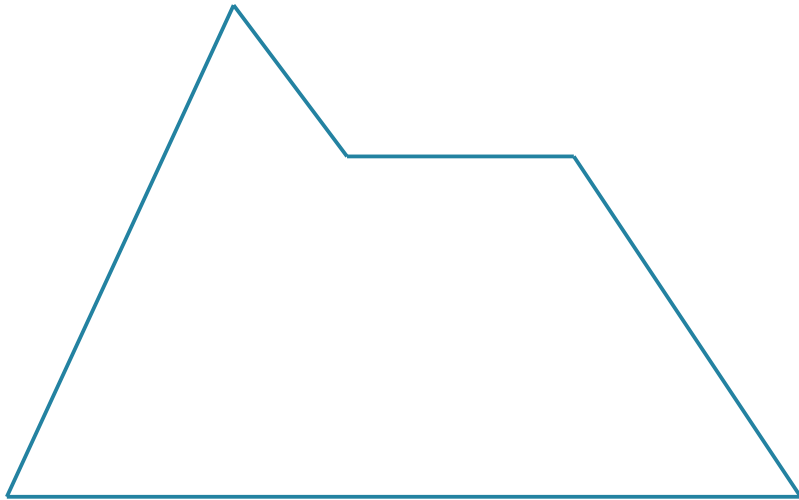
# Рептилии

- Тримино

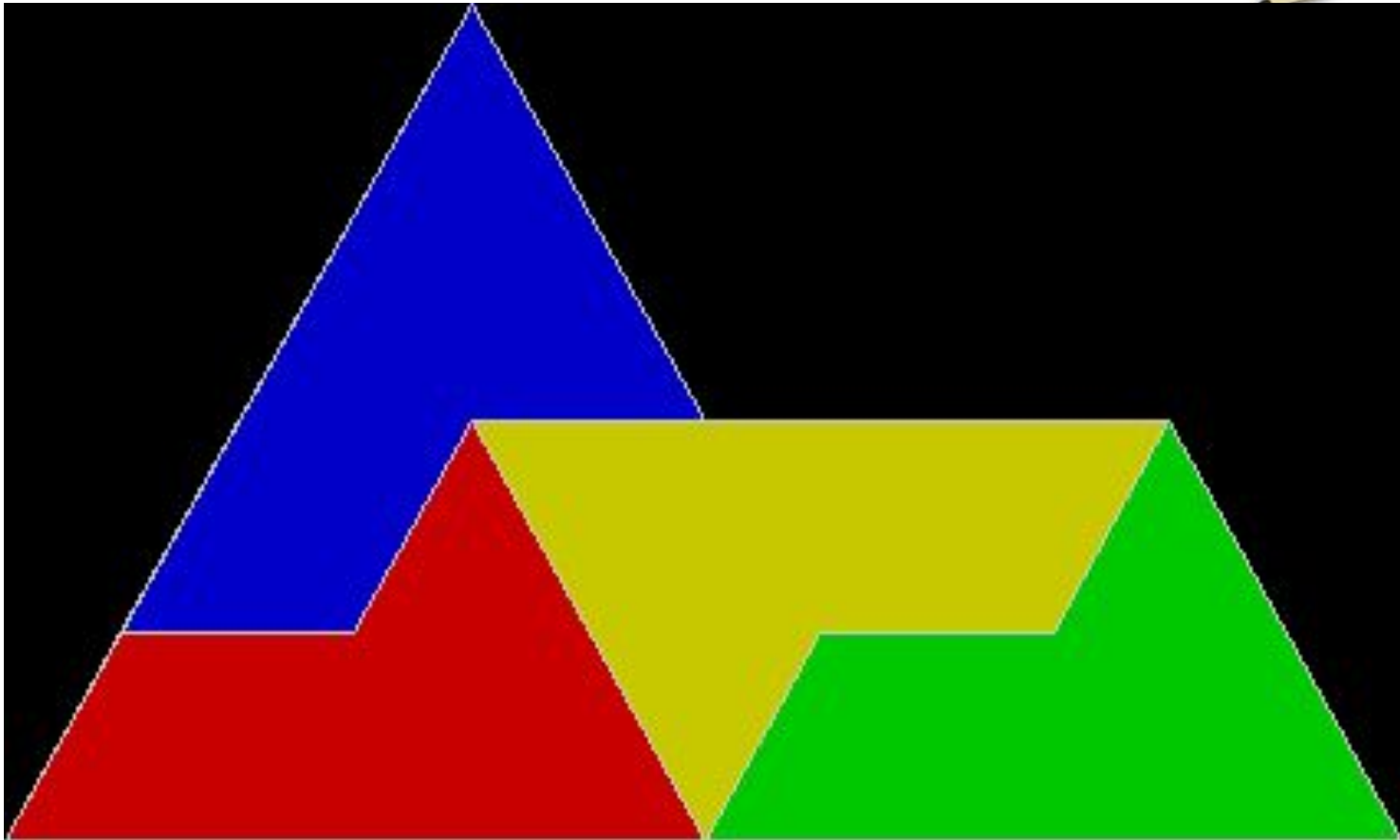


# Рептилии

- Сфинкс (как разместить внутри 4 меньших сфинкса?)



# Рептилии



# Мозаика Пенроуза



- Мозаика Пенроуза — общее название трёх особых типов непериодического разбиения плоскости; названы по имени английского математика Роджера Пенроуза, исследовавшего их в 1970-е годы.
- Все три типа, как и любые апериодические мозаики, обладают следующими свойствами:
- непериодичность — отсутствие трансляционной симметрии,
- повторяемость — любой сколь угодно большой фрагмент мозаики Пенроуза встречается в мозаике бесконечное число раз, хоть и через неравные расстояния,



# Мозаика Пенроуза

