

НЕРАВНОВЕСНАЯ АГРЕГАЦИЯ, ФРАКТАЛЫ

- Виноградова Варвара НФИбд-01-18
- Жижченко Глеб НФИбд-01-18
- Жижченко (Ветошкина) Валерия
НФИбд-03-18
- Греков Максим НФИбд-01-18
- Кондратьева Анастасия НФИбд-01-18
- Иванова Ольга НФИбд-01-18

Тема №1, Этап 2



Дерево пифагора

Для построения Дерева Пифагора используется генератор в виде прямоугольного треугольника с квадратом на гипотенузе. Применяя этот генератор к самому себе, получим первую итерацию с двумя новыми треугольниками.

Продолжая процесс, получим новые поколения, для которых количество треугольников увеличивается каждый в два раза. В пределе этого процесса «вырастет» Дерево Пифагора

Одним из свойств дерева Пифагора является то, что, если площадь первого квадрата равна единице, то на каждом уровне сумма площадей квадратов тоже будет равна единице.



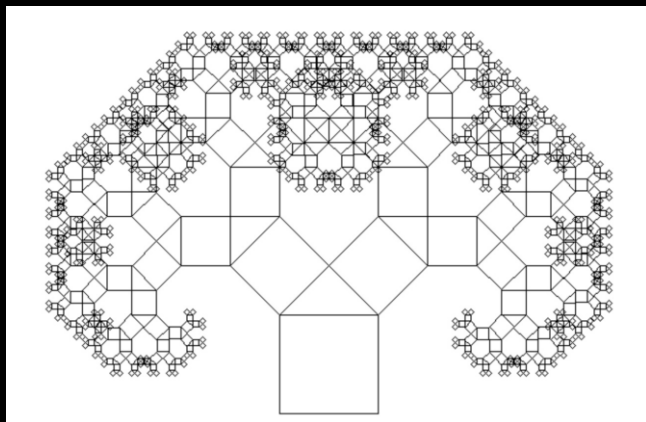


Рисунок 1. Классическое

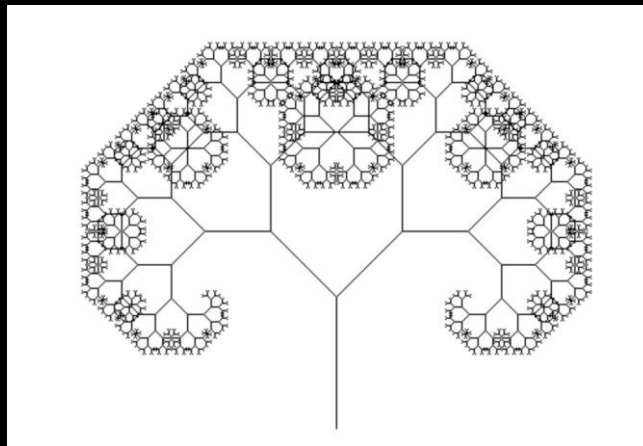


Рисунок 2. Обнаженное

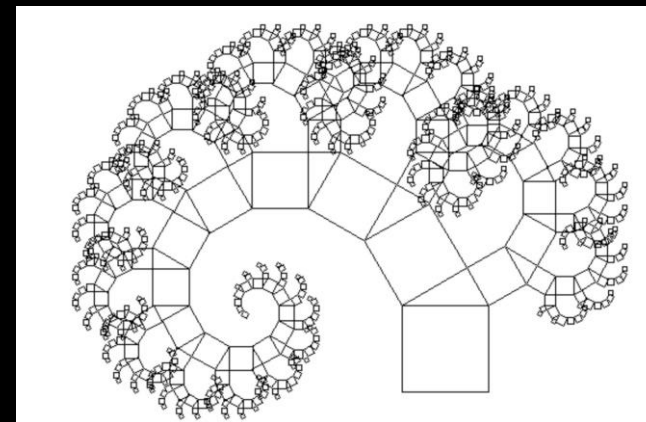


Рисунок 3. Обдуваемое ветром

Разновидности дерева

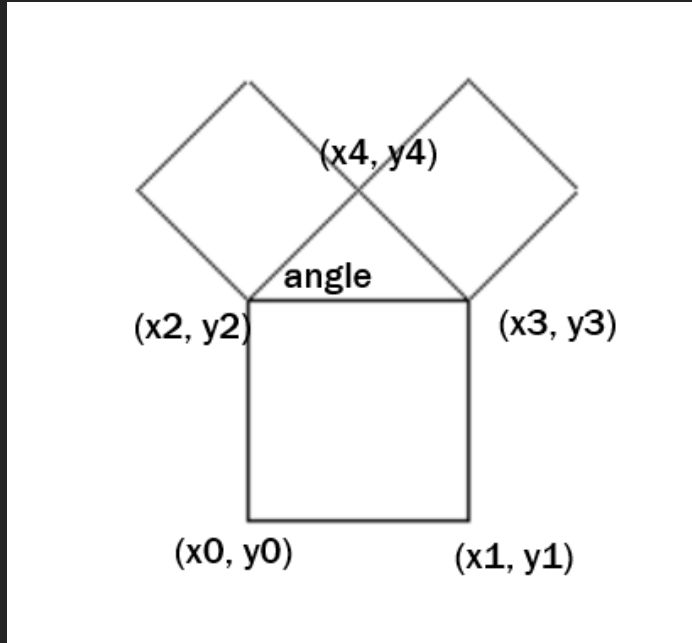
Как чертить фракталы при помощи Python?

Как правило, отрисовка фракталов сложна, так как глубинная природа фракталов определяется концепцией рекурсии. Говоря о графиках и их вычерчивании, мы обычно считаем, что они образованы пикселями или векторами, но количество пикселей или векторов всегда ограничено, а фракталы по определению бесконечно рекурсивны. Таким образом, попытавшись нанести фрактал на координатную сетку, мы в какой-то момент должны будем остановиться, и именно поэтому мы в данном случае говорим об «итерациях». На каждой итерации фрактал становится все сложнее, и в какой-то момент становится невозможно отличить две его итерации, следующие друг за другом (такой момент наступает, когда изменения происходят на уровне, сравнимом с размером пикселя). Здесь логично остановиться, но, как правило, форма фрактала вырисовывается быстрее, и остановиться можно еще раньше.

Входные данные

- Точка нижний левый угол
 - Длина стороны (length)
- Угол наклона основания (base_angle)
 - Угол наклона левой ветви (angle)

Задание точек



$$x1 = x0 + \text{length} * \cos(\text{base_angle})$$

$$y1 = y0 - \text{length} * \sin(\text{base_angle})$$

$$x2 = x0 + \text{length} * \cos(\text{base_angle} + 90)$$

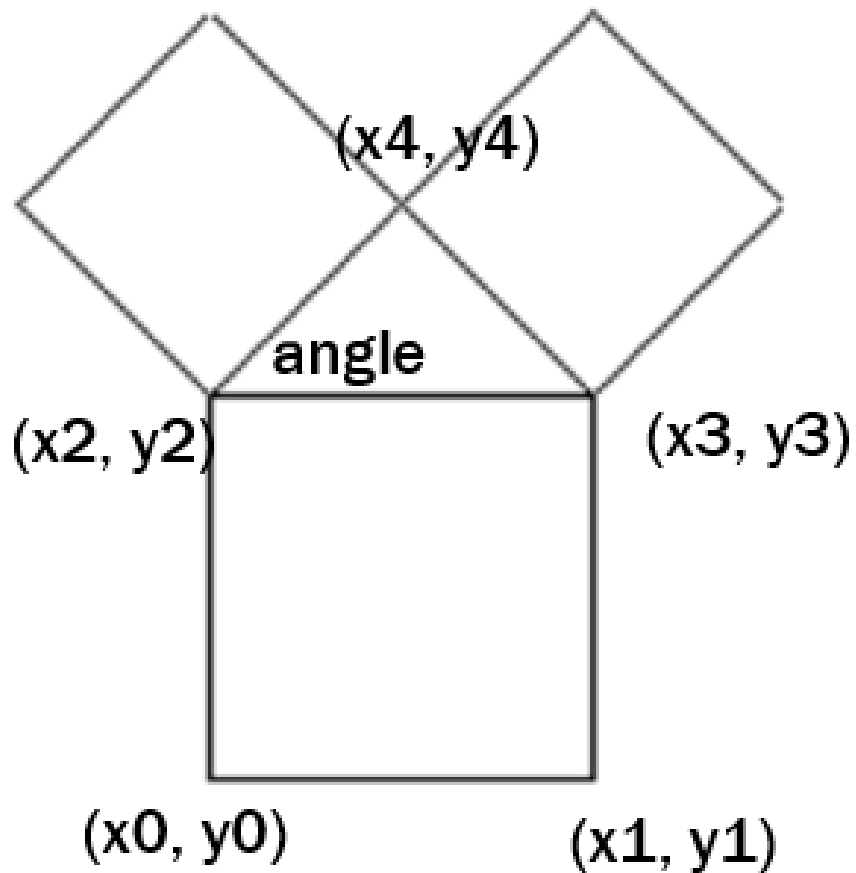
$$y2 = y0 - \text{length} * \sin(\text{base_angle} + 90)$$

$$x3 = x1 + \text{length} * \cos(\text{base_angle} + 90)$$

$$y3 = y1 - \text{length} * \sin(\text{base_angle} + 90)$$

$$x4 = x2 + \text{length} * \cos(\text{angle}) * \cos(\text{base_angle} + \text{angle})$$

$$y4 = y2 - \text{length} * \cos(\text{angle}) * \sin(\text{base_angle} + \text{angle})$$



Алгоритм прохождения итераций

Каждая следующая итерация начинается с нижнего левого угла, после чего мы находим все точки и рисуем их.

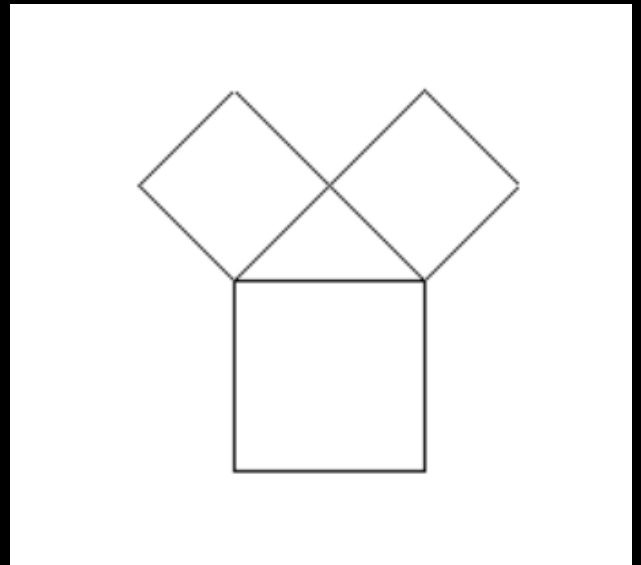
Затем соединяем точки между собой, получая линии.

Отрисовка левой ветви

Нижняя левая точка для левой ветви совпадает с (x_2, y_2) .

Длина нового левого дочернего квадрата равна $left_length = length * \cos(angle)$.

$base_angle$ для левого квадрата $base_angle$ родительского + $angle$



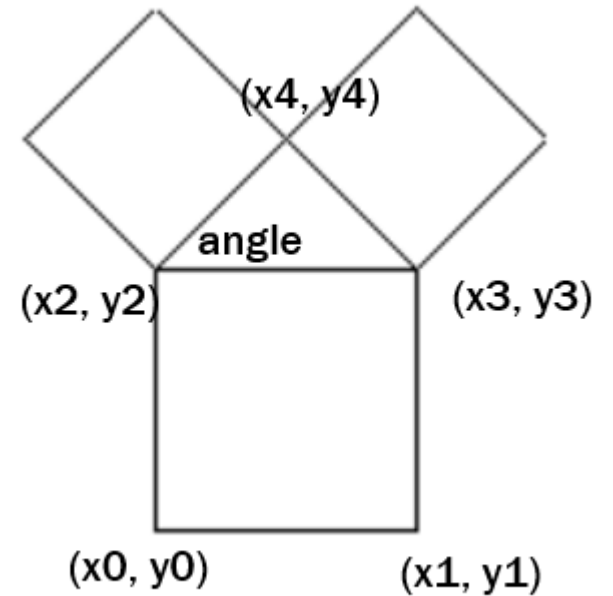
Отрисовка правой ветви

Нижняя левая точка для правого ветви совпадает с (x_4, y_4) .

Длина нового правого дочернего квадрата равна $right_length = length * \sin(angle)$.

$base_angle$ для правого квадрата

$base_angle$ родительского - $90^\circ + angle$



ВЫВОДЫ



Выбрали вид дерева для моделирования



Разобрались с этапами моделирования



Написали уравнения, которые будут использованы в коде