

```
[minimum width=minimum height=anchor=center] (blocktitle)at (, +); [minimum width=minimum height=anchor=center] (blockbody)at (, -0.5);
[line width=rounded corners=0] [draw=none](blockbody.south west) rectangle (blocktitle.north east); [color=blocktitlebgcolor, loosely dashed] (blocktitle.south west) – (blocktitle.south east);
[text width=, inner sep=anchor=center]at (blocktitle)
```

Исследование посвящено разработке алгоритма построения фрактала «Кривая Коха» на языке Python.

(blockbody) Рассмотрены математические основы и практическая реализация с использованием рекурсивных методов.

Ключевые слова: фракталы, кривая Коха, Python, рекурсия, геометрические преобразования.

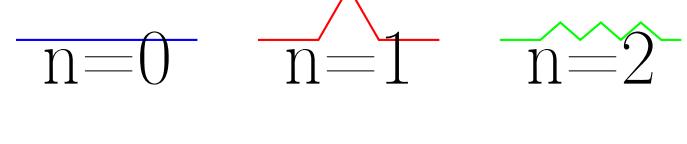
```
let 1 = (blockbody.south) in node ;
```

```
[minimum width=minimum height=anchor=center] (blocktitle)at (, +); [minimum width=minimum height=anchor=center] (blockbody)at (, -0.5);
[line width=rounded corners=0] [draw=none](blockbody.south west) rectangle (blocktitle.north east); [color=blocktitlebgcolor, loosely dashed] (blocktitle.south west) – (blocktitle.south east);
[text width=, inner sep=anchor=center]at (blocktitle)
```

Фракталы — геометрические объекты с бесконечно самоподобной структурой, сохраняющейся при любом масштабе.

(blockbody) Кривая Коха — классический фрактал, описанный шведским математиком Хельге фон Кохом в 1904 году.

Является непрерывной, но нигде не дифференцируемой кривой.

 Итерационное построение кривой Коха

```
let 1 = (blockbody.south) in node ;
```

```
[minimum width=minimum height=anchor=center] (blocktitle)at (, +); [minimum width=minimum height=anchor=center] (blockbody)at (, -0.5);
[line width=rounded corners=0] [draw=none](blockbody.south west) rectangle (blocktitle.north east); [color=blocktitlebgcolor, loosely dashed] (blocktitle.south west) – (blocktitle.south east);
[text width=, inner sep=anchor=center]at (blocktitle)
```

Основные формулы:

Длина кривой на n -ой итерации:

$$L_n = L_0 \times \left(\frac{4}{3}\right)^n$$

Количество элементарных отрезков:

$$N_n = 4^n$$

```
[minimum width=minimum height=anchor=center] (blocktitle)at (, +); [minimum width=minimum height=anchor=center] (blockbody)at (, -0.5);
[line width=rounded corners=0] [draw=none](blockbody.south west) rectangle (blocktitle.north east); [color=blocktitlebgcolor, loosely dashed] (blocktitle.south west) – (blocktitle.south east);
[text width=, inner sep=anchor=center]at (blocktitle)
```

Фрактальная размерность (размерность Хаусдорфа):

$$D = \frac{\log 4}{\log 3} \approx 1.26186$$

Геометрическое построение: Каждый отрезок делится на 3 равные части, средняя часть заменяется на два отрезка длиной $1/3$, образующих равносторонний треугольник.

```
let 1 = (blockbody.south) in node ;
```

```
[minimum width=minimum height=anchor=center] (blocktitle)at (, +); [minimum width=minimum height=anchor=center] (blockbody)at (, -0.5);
[line width=rounded corners=0] [draw=none](blockbody.south west) rectangle (blocktitle.north east); [color=blocktitlebgcolor, loosely dashed] (blocktitle.south west) – (blocktitle.south east);
[text width=, inner sep=anchor=center]at (blocktitle)
```

Рекурсивный подход:

```
def koch_curve(t, length, depth) :
    """t — length — depth"""
    if depth == 0 :
        t.forward(length)
    else :
        3*length/
        = 3.04koch_curve(t, length, depth - 1).left(60)60koch_curve(t, length, depth - 1).
        1).right(120)120koch_curve(t, length, depth - 1).left(60)60koch_curve(t, length, depth - 1)
def koch_snowflake(t, length, depth) : for i in range(3) : koch_curve(t, length, depth).right(120)
```

```
let 1 = (blockbody.south) in node ;
```

```
[minimum width=minimum height=anchor=center] (blocktitle)at (, +); [minimum width=minimum height=anchor=center] (blockbody)at (, -0.5);
[line width=rounded corners=0] [draw=none](blockbody.south west) rectangle (blocktitle.north east); [color=blocktitlebgcolor, loosely dashed] (blocktitle.south west) – (blocktitle.south east);
[text width=, inner sep=anchor=center]at (blocktitle)
```

Инструменты и технологии:

- Python 3.8+
- Turtle Graphics для визуализации
- Matplotlib для анализа данных
- Time module для измерения производительности

(blockbody) Методика исследования:

- Анализ рекурсивных алгоритмов
- Исследование вычислительной сложности
- Визуализация геометрических преобразований
- Сравнение теоретических и практических результатов

```
let 1 = (blockbody.south) in node ;
```

```
[minimum width=minimum height=anchor=center] (blocktitle)at (, +); [minimum width=minimum height=anchor=center] (blockbody)at (, -0.5);
[line width=rounded corners=0] [draw=none](blockbody.south west) rectangle (blocktitle.north east); [color=blocktitlebgcolor, loosely dashed] (blocktitle.south west) – (blocktitle.south east);
[text width=, inner sep=anchor=center]at (blocktitle)
```

Численные характеристики:

n	Отрезки	Длина	Время (с)
0	1	1.00	0.001
1	4	1.33	0.005
2	16	1.78	0.021
3	64	2.37	0.085
4	256	3.16	0.341
5	1024	4.21	1.364

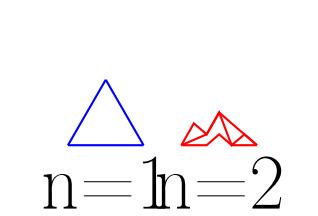
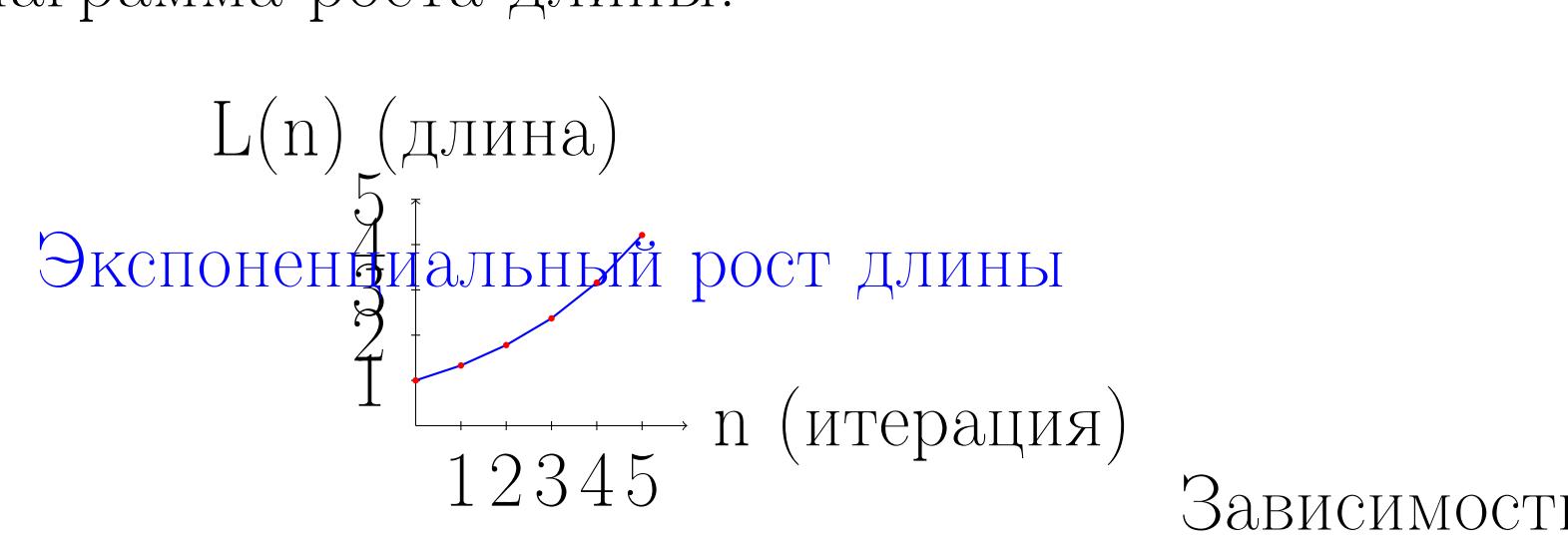
(blockbody)  Снежинка Коха на разных итерациях

Диаграмма роста длины:



длины кривой от номера итерации

```
let 1 = (blockbody.south) in node ;
```

```
[minimum width=minimum height=anchor=center] (blocktitle)at (, +); [minimum width=minimum height=anchor=center] (blockbody)at (, -0.5);
[line width=rounded corners=0] [draw=none](blockbody.south west) rectangle (blocktitle.north east); [color=blocktitlebgcolor, loosely dashed] (blocktitle.south west) – (blocktitle.south east);
[text width=, inner sep=anchor=center]at (blocktitle)
```

Основные результаты:

- Успешно реализован рекурсивный алгоритм построения кривой Коха
- Исследованы фрактальные свойства кривой (самоподобие, размерность)
- Проанализирована вычислительная сложность алгоритма: $O(4^n)$
- (blockbody) • Подтверждены теоретические расчеты длины кривой

Практическое применение:

- Образовательные цели в математике и информатике
- Компьютерная графика и моделирование природных объектов
- Исследование рекурсивных алгоритмов
- Основы фрактальной геометрии

let 1 = (blockbody.south) in node ;

[minimum width=minimum height=anchor=center] (blocktitle)at (, +); [minimum width=minimum height=anchor=center] (blockbody)at (, -0.5);

[line width=rounded corners=0] [draw=none][blockbody.south west] rectangle (blocktitle.north east); [color=blocktitlebgcolor, loosely dashed] (blocktitle.south west) – (blocktitle.south east);
[text width=, inner sep=anchor=center]at (blocktitle) ; [text width=, inner sep=anchor=center]at

1. Mandelbrot B. B. The Fractal Geometry of Nature. — 1982.

2. Peitgen H.-O., Jürgens H., Saupe D. Chaos and Fractals. — 1992.

(blockbody) 3. Федер Е. Фракталы. — М.: Мир, 1991.

4. Python Documentation. — 2024.

5. Кроновер Р. М. Фракталы и хаос в динамических системах. — 2000.

let 1 = (blockbody.south) in node ;

