

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

Національний авіаційний університет

Навчально-науковий інститут інформаційно-діагностичних систем

Кафедра прикладної математики

**Звіт по лабораторній роботі №2.2**

**«Побудова фракталів»**

Виконав:

Студент 351 групи ІДС

Тітенко Д.С.

Викладач: Юрчук І. А.

Київ 2016

**Мета:** Навчитися будувати різнокольорові конструктивні фрактали за допомогою генератора.

### **Завдання:**

- 1.Провести аналіз конструктивного фракталу, що представлений своїм генератором ,за допомогою масштабування та повороту.
- 2.Реалізувати програмно алгоритми побудови фракталу за заданим генератором G, що зображений у Додатку1,та числом ітерацій  $p=3$ .

### **Варіант 19(Буква М)**

### **Теоретичні відомості**

Поняття фрактал, фрактальна геометрія та фрактальна графіка, з'явилися в кінці 70-х, сьогодні міцно увійшли в ужиток математиків і комп'ютерних художників. Слово фрактал утворене від латинського *fractus* і в перекладі означає «складається з фрагментів». Воно було запропоновано математиком, Бенуа Мандельброт в 1975 році для позначення нерегулярних, але самоподібних структур, якими він займався. У роботах Мандельброта використані наукові результати інших вчених, які працювали в 1875 - 1925 роках в тій же області (Пуанкаре, Фату, Жюліа, Кантор, Хаусдорф). Але тільки в наш час вдалося об'єднати їх роботи в єдину систему.

Фракталом (лат. *fractus* - Подрібнений, зламаний, розбитий) називається структура, що складається з частин, які в якомусь сенсі подібні цілому.

З усіх типів фракталів найбільш наочними є геометричні фрактали. У двовірному випадку їх отримують за допомогою деякої ламаної, в тривірному випадку - поверхні, званої генератором. За один крок алгоритму кожен з відрізків, що складають ламану, замінюється на ламану-генератор у відповідному масштабі. В результаті нескінченного повторення цієї процедури виходить геометричний фрактал.

Одним з основних властивостей фракталів є самоподібність. Об'єкт називають самоподібним, коли збільшені частини об'єкта схожі на сам об'єкт і один на одного.

Перефразовуючи це визначення, можна сказати, що в найпростішому випадку невелика частина фрактала містить інформацію про все фрактале.

Найпростішим фрактальним об'єктом є фрактальний трикутник. Спробуємо побудувати звичайний рівносторонній трикутник зі стороною  $a$ . Розділимо

кожну з його сторін на три відрізка. На середньому відрізку сторони побудуємо рівносторонній трикутник зі стороною, рівною  $1/3$  сторони початкового трикутника, а на інших відрізках побудуємо рівносторонній трикутники зі стороною, рівною  $(1/9) \cdot a$ . З отриманими трикутниками повторимо ті ж операції. Можна помітити, що трикутники наступних поколінь успадковують властивості своїх батьківських структур. Так народжується фрактальна фігура.

Процес спадкування можна продовжувати до нескінченності. Взявши такий нескінченний фрактальний об'єкт і розглянувши його в лупу або мікроскоп, можна знайти в ньому все нові і нові деталі, що повторюють властивості вихідної структури.

Фрактальні властивості мають багато об'єктів живої і неживої природи. Звичайна сніжинка, багаторазово збільшена, виявляється фрактальним об'єктом. Фрактальні алгоритми лежать в основі росту кристалів і рослин. Погляньте на гілку папоротевого рослини, і ви побачите, що кожна дочірня гілка багато в чому повторює властивості гілки більш високого рівня.

В окремих гілках дерев чисто математичними методами можна простежити властивості всього дерева. А якщо гілку поставити в воду, то незабаром можна отримати саджанець, який з часом розвинеться в повноцінне дерево.

Ця ідея знайшла використання в комп'ютерній графіці завдяки компактності математичного апарату, необхідного для її реалізації. Фрактальна графіка, як і векторна - обчислюється, але відрізняється від неї тим, що ніякі об'єкти в пам'яті комп'ютера не зберігаються. Зображення будується за рівнянням (або по системі рівнянь), тому нічого, крім формули, зберігати не треба. Зміна коефіцієнтів в рівнянні дозволяє отримувати абсолютно іншу картину. Так, за допомогою декількох математичних коефіцієнтів можна задати лінії і поверхні дуже складної форми. Сьогодні розроблені алгоритми синтезу коефіцієнтів фрактала, що дозволяє відтворити копію будь-якої картини, як завгодно близькою до вихідного оригіналу. Фактично завдяки фрактальній графіці знайдений спосіб ефективної реалізації складних неевклідових об'єктів, образи яких дуже схожі на природні.

Геометричні фрактали на екрані комп'ютера - це візерунки, побудовані самим комп'ютером за заданою програмою (рисунок 9.4). Вони дуже красиві, незвичайні і цікаві. Багато художників на Заході (наприклад, Мелісса Бінде) розглядають фрактали як новий вид комп'ютерного мистецтва. Крім фрактальної живопису існують фрактальна анімація і фрактальна музика.

Здатність фрактальної графіки моделювати образи живої природи обчислювальним шляхом часто використовують для автоматичної генерації незвичайних ілюстрацій.

Програмні засоби для роботи з фрактальною графікою призначені для автоматичної генерації зображень шляхом математичних розрахунків. Створення художньої фрактальної композиції полягає не в малюванні або оформленні, а в програмуванні. Фрактальну графіку рідко застосовують для створення друкованих або електронних документів, але її часто використовують у розважальних програмах.

Таким чином, фрактал це об'єкт, окремі елементи якого успадковують властивості батьківських структур. Оскільки більш детальний опис елементів меншого масштабу відбувається по простому алгоритму, описати такий об'єкт можна всього лише декількома математичними рівняннями. Фрактали дозволяють описувати цілі класи зображень, для детального опису яких потрібно відносно мало пам'яті. З іншого боку, фрактали слабо застосовні до зображень поза цими класів.

## Код програми

```
from tkinter import *
from math import *
import sys

SIZE_X = 1080
SIZE_Y = 1240
STEP = 300
d = 3

class Turtle:
    x = 0
    y = 0
    a = 0
    pen = 1
    canvas = None

    def __init__(self, canvas):
        self.canvas = canvas

    def setpen(self, yes):
        self.pen = yes

    def rotate(self, g):
        self.a += g

    def move(self, sx, sy):
        self.x = sx
        self.y = sy

    def forward(self, length):
        new_x = self.x + cos(pi/180*self.a)*length
        new_y = self.y + sin(pi/180*self.a)*length
        cx = SIZE_X / 2
        cy = SIZE_Y / 2
        if self.pen:
            self.canvas.create_line(cx-self.x, cy-self.y, cx-new_x, cy-new_y, fill="white")
        self.x = new_x
        self.y = new_y

    def line(self, angle, length):
        new_x = self.x + cos(pi/180*angle)*length
        new_y = self.y + sin(pi/180*angle)*length
        cx = SIZE_X / 2
        cy = SIZE_Y / 2
        if self.pen:
            self.canvas.create_line(cx-self.x, cy-self.y, cx-new_x, cy-new_y, fill="white")
        self.x = new_x
        self.y = new_y
```

```
class Fractal:
```

```
    turtle = None
```

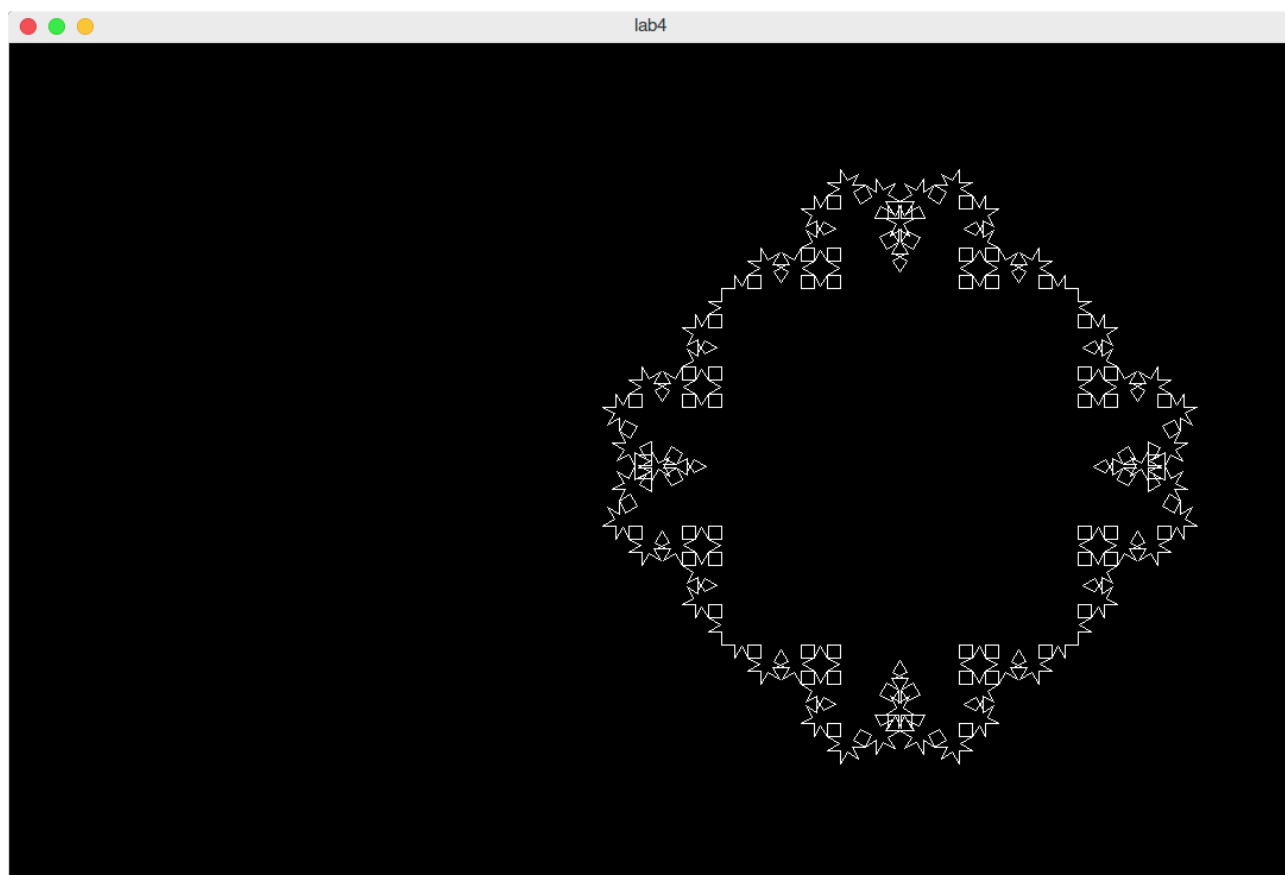
```
    def __init__(self,turtle):  
        self.turtle = turtle
```

```
    def make(self,level):  
        self.turtle.move(-SIZE_X / 3, SIZE_Y / 3)  
        self.f(STEP, level)  
        self.turtle.rotate(-90)  
        self.f(STEP, level)  
        self.turtle.rotate(-90)  
        self.f(STEP, level)  
        self.turtle.rotate(-90)  
        self.f(STEP, level)
```

```
    def f(self, length, depth):  
        if depth == 0:  
            self.turtle.forward(length)  
        else:  
            self.f(length / 3, depth - 1)  
            self.turtle.rotate(90)  
            self.f(length / 3, depth - 1)  
            self.turtle.rotate(-150)  
            self.f(length / 3, depth - 1)  
            self.turtle.rotate(120)  
            self.f(length / 3, depth - 1)  
            self.turtle.rotate(-150)  
            self.f(length / 3, depth - 1)  
            self.turtle.rotate(90)  
            self.f(length / 3, depth - 1)
```

```
if __name__ == '__main__':  
    root = Tk()  
    root.title("lab4")  
    c = Canvas(root, width=SIZE_X, height=SIZE_Y, bg="black")  
    c.pack()  
    fractal = Fractal(Turtle(c))  
    fractal.make(d)  
    root.mainloop()
```

## Результати роботи програми



## **Висновки**

Я навчився будувати та аналізувати конструктивний фрактал, представлений генератором, використовуючи масштабування та поворот