



СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“  
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

## КУРСОВ ПРОЕКТ ПО ФРАКТАЛИ

Тема:

Триъгълник на Серпински

Студент:

Семир Балджиев, 8MI0700024

София 2024 г.

# 1. Триъгълник на Серпински. Въведение

Триъгълникът на Серпински, наричан още и Решето на Серпински, е фрактал с формата на равностранен триъгълник, рекурсивно разделен на по-малки равностранни триъгълници с дължина на страната, съответно  $1/2$ ,  $1/4$ ,  $1/8$ , ...  $1/2^n$  ( $n \rightarrow \infty$ ) от дължината на външния триъгълник. Триъгълникът на Серпински е един от най-простите примери за самоподобни множества, тъй като е математически генериран модел, който може да се самовъзпроизведе при произволно увеличаване или намаляване на мащаба. Наречен е на името на полския математик Вацлав Серпински, но съществува от много столетия като декоративен елемент, преди Серпински да започне да изучава свойствата му на математически обект.

## Конструкции

Има много различни начини за конструиране на триъгълника на Серпински и някои от тях са:

### Премахване на триъгълници

Триъгълникът на Серпински може да бъде конструиран от равностранен триъгълник чрез многократно премахване на триъгълни подмножества:

1. Започнете с равностранен триъгълник.
2. Разделете го на четири по-малки еднакви равностранни триъгълника и премахнете централния триъгълник.
3. Повторете стъпка 2 с всеки от останалите по-малки триъгълници безкрайно.



Фиг. 1. Конструиране чрез премахване на триъгълници

## Свиване и дублиране

Същата последователност от фигури, събиращи се към триъгълника на Серпински, може алтернативно да бъде генерирана чрез следните стъпки:

Започнете с произволен триъгълник в равнина (всяка затворена, ограничена област в равнината всъщност ще работи). Каноничният триъгълник на Серпински използва равностранен триъгълник с основа, успоредна на хоризонталната ос

Свийте триъгълника до  $\frac{1}{2}$  височина и  $\frac{1}{2}$  ширина, направете три копия и позиционирайте трите свити триъгълника така, че всеки триъгълник да докосва другите два триъгълника в ъгъл. Обърнете внимание на появата на централната дупка - защото трите свити триъгълника могат да покриват само между тях  $\frac{3}{4}$  от площта на оригинала. (Дупките са важна характеристика на триъгълника на Серпински.)

Повторете стъпка 2 с всеки от по-малките триъгълници.

Този безкраен процес не зависи от това, че началната форма е триъгълник - просто така е по-ясно. Първите няколко стъпки, започващи например от квадрат, също клонят към триъгълник на Серпински.



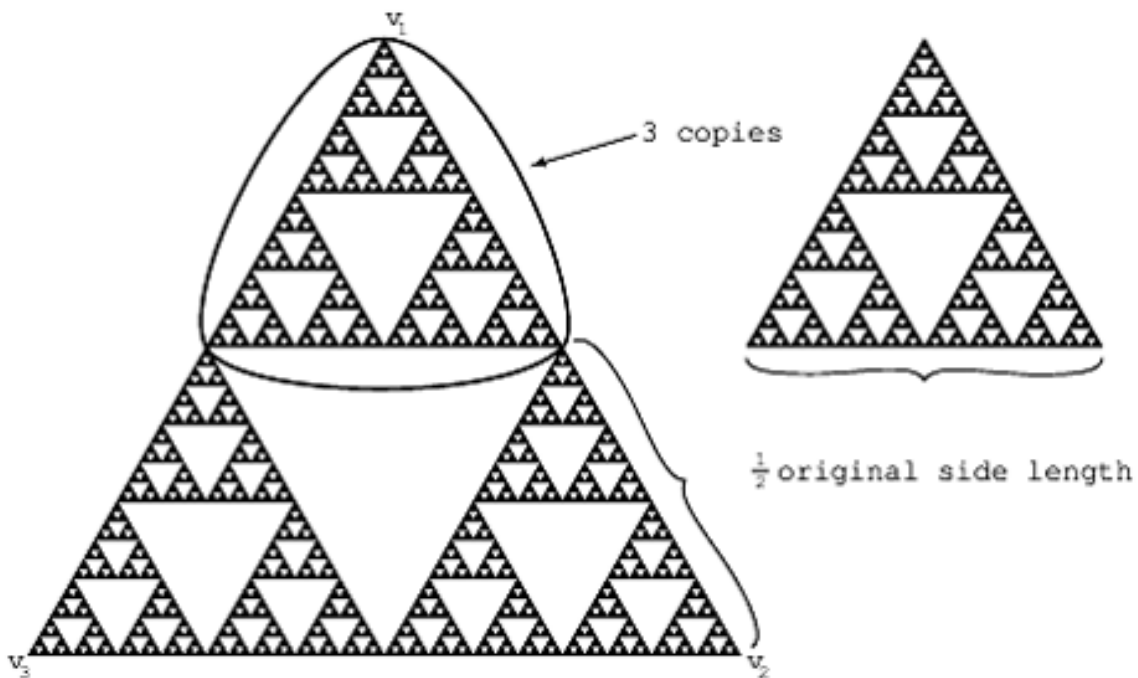
Фиг. 2. Конструирание чрез свиване и дублиране

## 3. Свойства

За целочислен брой измерения  $d$  когато удвоите страна на обект, се създават  $2^d$  негови копия т.е. 2 копия за 1-измерен обект, 4 копия за 2-измерен обект и 8 копия за 3-измерен обект. За триъгълника на Серпински, удвояването на неговата страна създава 3 негови копия. Така триъгълникът на Серпински има Хаусдорфова размерност:  $\frac{\log 3}{\log 2} \approx 1.585$  което следва от решаването на  $2^d = 3$  за  $d$ .

Площта на триъгълник на Серпински е нула (в мярка на Лебег). Оставащата площ след всяка итерация е  $\frac{3}{4}$  от областта от предишната итерация и безкраен брой итерации води до област, приближаваща се до нула.

Точките на триъгълника на Серпински имат проста характеристика в барицентрични координати. Ако една точка има барицентрични координати  $(0.u_1u_2u_3..., 0.v_1v_2v_3..., 0.w_1w_2w_3...)$  изразени като двоични числа, тогава точката е в триъгълника на Серпински тогава и само ако  $u_i + v_i + w_i = 1$  за всяко  $i$ .



Фиг. 3. Триъгълник на Серпински с описани характеристики

## 4. Програмна реализация. Описание на кода

За програмната реализация е използван езика Python. Кодът използва модула Turtle в Python за рисуване на Триъгълника на Серпински рекурсивно.

За тази цел се използва функцията `draw_sierpinski_triangle`, която рекурсивно рисува триъгълници във вътрешността на други триъгълници, като намалява мащаба им, докато не се достигне зададената редица.

Като функцията `draw_sierpinski_triangle` преминава през поредица от рекурсивни извиквания, всеки път рисува с по-малък размер, това създава впечатляващ ефект на "фрактал". Всяко ниво на рекурсия съдържа три триъгълника, като центърът на всяка страна се използва за рекурсивните извиквания. Триъгълниците са оцветени в червен цвят, който се може да се променя като се задава на функцията `fillcolor` от модула Turtle.

```
def draw_sierpinski_triangle(turtle, order, size):
    if order == 0:
        turtle.fillcolor("red")
        turtle.begin_fill()
        for _ in range(3):
            turtle.forward(size)
            turtle.left(120)
        turtle.end_fill()
    else:
        size /= 2
        draw_sierpinski_triangle(turtle, order - 1, size)
        turtle.forward(size)
        draw_sierpinski_triangle(turtle, order - 1, size)
        turtle.backward(size)
        turtle.left(60)
        turtle.forward(size)
        turtle.right(60)
        draw_sierpinski_triangle(turtle, order - 1, size)
        turtle.left(60)
        turtle.backward(size)
        turtle.right(60)
```

Фиг. 4. Рекурсивна функция за рисуване на фрактала

Входните параметри на функцията са turtle (обектът, който използваме за рисуване с Turtle), order (редицата на триъгълника, която определя колко пъти ще се извика рекурсията) и size (размерът на текущия триъгълник).

В началото на функцията проверяваме дали order е 0. Ако е 0, това означава, че сме достигнали основния случай на рекурсията и трябва да нарисуваме триъгълник. В този случай функцията просто рисува триъгълник с дадения размер и цвят.

Ако order не е 0, значи сме в рекурсивен случай. Тогава намаля размера на триъгълника наполовина ( $size /= 2$ ) и извиква функцията draw\_sierpinski\_triangle рекурсивно три пъти за да нарисуваме по-малките триъгълници. По същия начин, като нарисуваме триъгълници в триъгълници, продължаваме рекурсивно да намалява размера на триъгълника и да го разполагаме правилно, докато не достигнем основния случай.

Ключът за разбиране на рекурсивната функция draw\_sierpinski\_triangle е, че при всяко извикване намалява редицата order с 1 и намалява размера на триъгълника наполовина, като този процес се повтаря рекурсивно докато не се достигне основния случай (order == 0). Тогава се рисува най-малкият триъгълник и рекурсията приключва.

Функцията main се грижи за настройката на прозореца и вземането на поредицата на триъгълника от потребителя, след което стартира рисуването на триъгълника. Прозореца заема почти цялата ширина и височина на екрана, а рисуването започва от долния ляв ъгъл на екрана.

```

def main():
    window = turtle.Screen()
    window.bgcolor("white")
    window.setup(width=0.99, height=0.9)
    sierpinski_turtle = turtle.Turtle()
    sierpinski_turtle.penup()
    sierpinski_turtle.setpos(-window.window_width() / 2, -window.window_height() / 2)
    sierpinski_turtle.pendown()
    sierpinski_turtle.speed(100)

    order = int(input("Enter the order of the Sierpiński triangle: "))
    size = order*100

    if size == 0: size = 100

    draw_sierpinski_triangle(sierpinski_turtle, order, size)

    window.exitonclick()

```

Фиг. 5. Основната функция на програмата

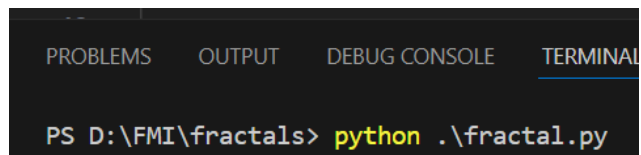
Това позволява на програмата да рисува триъгълници от всеки възможен размер и поредица, като се гарантира красив и въздействащ визуален резултат. Важно е да се отбележи, че за въвеждането на поредицата се предполага, че потребителят ще въведе положително цяло число.

## 5. Предварителни изисквания за стартиране на програмата

Програмата представлява конзолно приложение и приема едно положително число като вход (нивото на триъгълника).

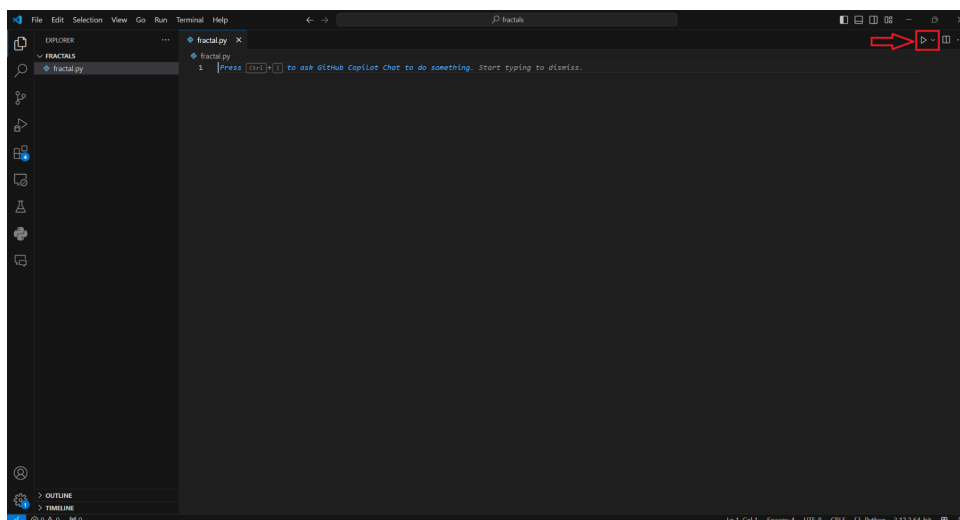
За да бъде изпълнена програмата трябва да се конфигурира средата за изпълнение и да се инсталират следните инструменти:

1. Инсталиран Python на съответния компютър.
  - а. Може да се изтегли от официалния сайт. Следват се стъпките от инсталатора.
2. Инсталирана среда за разработка (IDE) (VS Code/PyCharm)
3. Инсталиране на модула turtle чрез командата: `pip install turtle`
4. Изпълнение на програмата чрез команда от терминала или чрез бутона за стартиране
  - а. чрез команда: `python .\fractal.py`



Фиг. 6. Команда за изпълнение

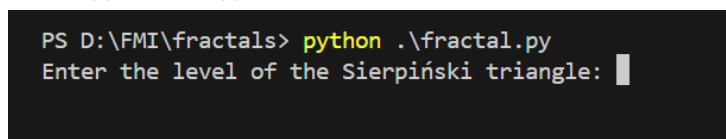
б. със следния бутон:



Фиг. 7. Бутон за изпълнение

## 6. Примерно изпълнение на програмата

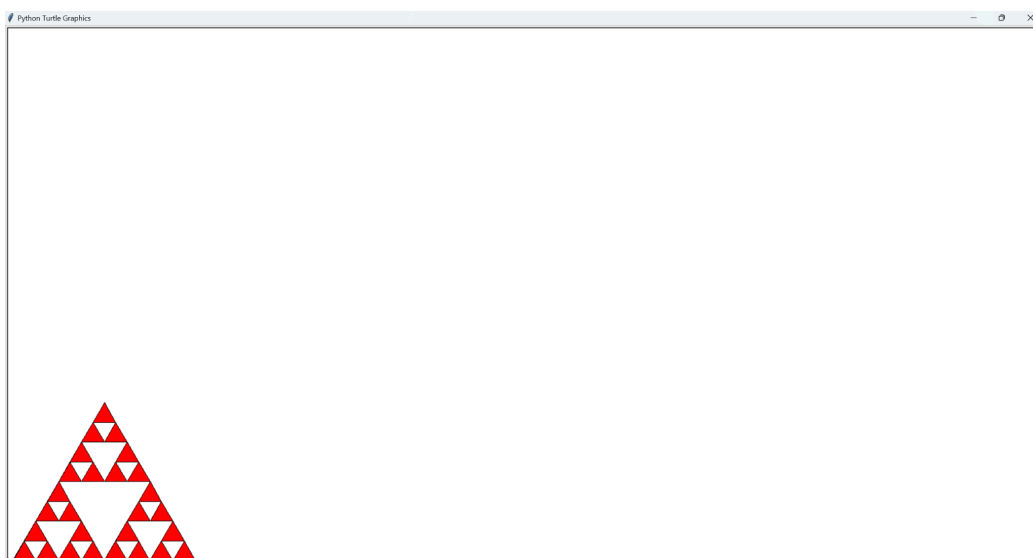
При стартиране на програмата се очаква вход от потребителя който да се въведе в конзолата която изглежда по следния начин:



Фиг. 8. Вход на ниво на триъгълника от потребителя

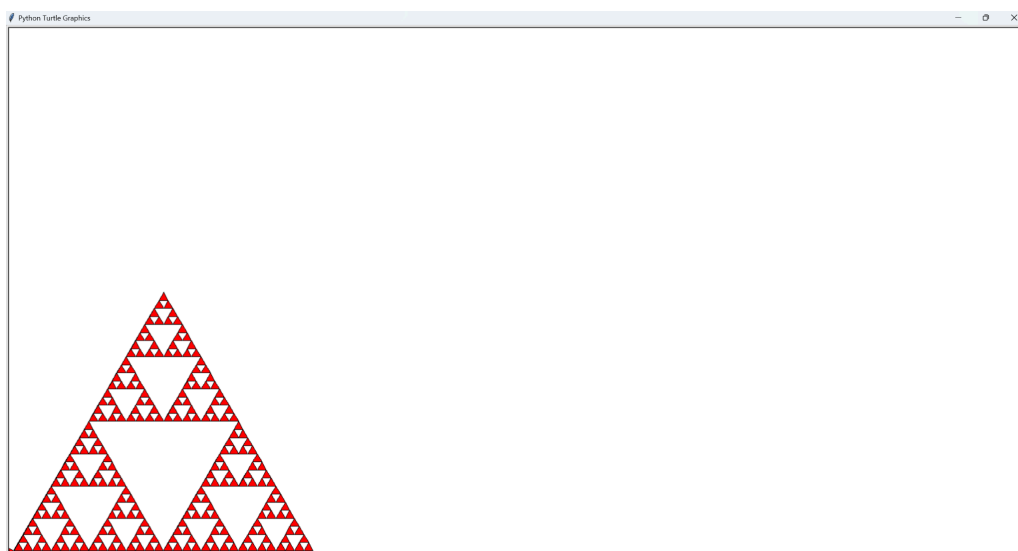
След като потребителят въведе желаното от него ниво се отваря прозорец и започва рисуването на Триъгълника на Серпински.

```
PS D:\FMI\fractals> python .\fractal.py
Enter the level of the Sierpiński triangle: 3
```



Фиг. 9. Триъгълник на Серпински с ниво 3

```
PS D:\FMI\fractals> python .\fractal.py
Enter the level of the Sierpiński triangle: 5
```



Фиг. 10. Триъгълник на Серпински с ниво 5

*Забележка: При по-голямо ниво отнема повече време за рисуване и триъгълниците са малки и не се виждат добре затова е препоръчително да се тества с ниво в диапазона (0-6)*