

Санкт-Петербургский Государственный  
Электротехнический Университет

Кафедра МОЭВМ

Задание для лабораторной работы № 3  
"П о с т р о е н и е ф р а к т а л о в "

Выполнили:

Доброхвалов М.О. 6303,

Черкасова Е.И. 6382

Факультет: ФКТИ

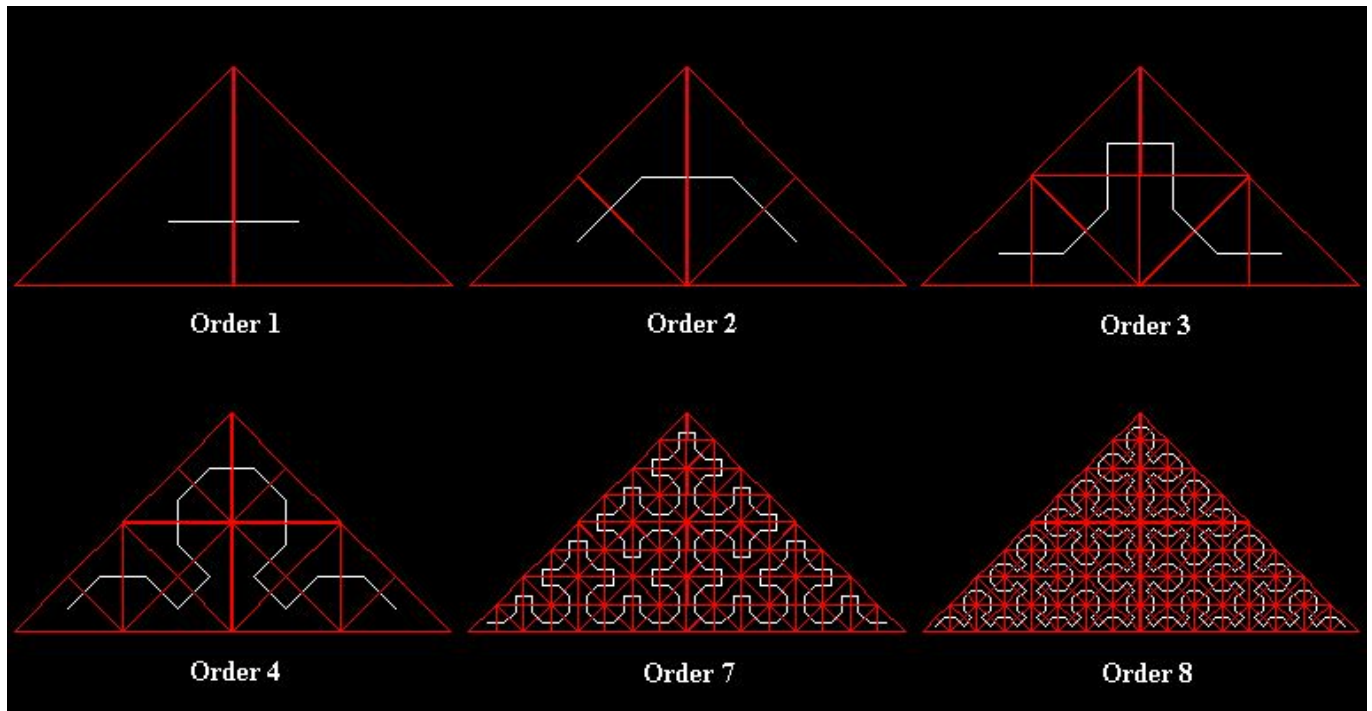
Преподаватель: Герасимова Т.В.

Санкт-Петербург

2019 г.

## Задание

На базе предыдущей лабораторной работы разработать программу реализующую фрактал по индивидуальному заданию.



## Общие сведения

Понятия фрактал и фрактальная геометрия, появившиеся в конце 70-х, с середины 80-х прочно вошли в обиход математиков и программистов. Слово фрактал образовано от латинского *fractus* и в переводе означает состоящий из фрагментов. Оно было предложено Бенуа Мандельбротом в 1975 году для обозначения нерегулярных, но самоподобных структур, которыми он занимался. Рождение фрактальной геометрии принято связывать с выходом в 1977 году книги Мандельброта 'The Fractal Geometry of Nature'. В его работах использованы научные результаты других ученых, работавших в период 1875-1925 годов в той же области (Пуанкаре, Фату, Жюлиа, Кантор, Хаусдорф). Но только в наше время удалось объединить их работы в единую систему.

Фрактал (лат. fractus — дробленный) — термин, означающий геометрическую фигуру, обладающую свойством самоподобия, то есть составленную из нескольких частей, каждая из которых подобна всей фигуре целиком.

Существует большое число математических объектов, называемых фракталами (треугольник Серпинского, снежинка Коха, кривая Пеано, множество Мандельброта). Фракталы с большой точностью описывают многие физические явления и образования реального мира: горы, облака, турбулентные (вихревые) течения, корни, ветви и листья деревьев, кровеносные сосуды, что далеко не соответствует простым геометрическим фигурам.

Классификация фракталов:

- Геометрические
- Алгебраические
- Стохастические

Способы построения фракталов:

- L-система

L-система (от имени Lindenmayer) - это грамматика некоторого языка (достаточно простого), которая описывает инициатор и преобразование, выполняемое над ним, при помощи средств, аналогичных средствам языка Лого (аксиоматическое описание простейших геометрических фигур и допустимых преобразований на плоскости и в пространстве).

- Система итерирующих функций IFC

Применение таких преобразований, которые дают ту фигуру которую необходимо. Система итерирующих функций - это совокупность сжимающих аффинных преобразований. Как известно, аффинные преобразования включают в

себя масштабирование, поворот и параллельный перенос. Аффинное преобразование считается сжимающим, если коэффициент масштабирования меньше единицы.

## Выполнение работы

Работа выполнена с помощью инструмента для разработки GUI приложений PyQt5, в основе которого лежит Python 3.6 и библиотеки OpenGL для Python.

Отображение фигуры фрактала осуществляется с помощью следующих функций:

```
def get_objects(basic_triangles, basic_lines, connections, depth):
    if depth != order:
        triangles = rotate(basic_triangles/2**(0.5), 3*np.pi/4)
        lines = rotate(basic_lines/2**(0.5), 3*np.pi/4)

        shift = triangles[3] - basic_triangles[0]
        triangles -= shift
        lines -= shift

        triangles_copy = np.matmul(triangles.copy(), mirror_matrix)
        lines_copy = np.matmul(lines.copy(), mirror_matrix)

        triangles = np.append(triangles, triangles_copy, axis=0)
        lines = np.append(lines, lines_copy, axis=0)

    if depth > 1:
        connections = rotate(connections/2**(0.5), 3*np.pi/4)
        connections -= shift
        connections_copy = np.matmul(connections.copy(), mirror_matrix)
        connections = np.append(connections, connections_copy, axis=0)
    point_index = 2**((depth-1) if (depth > 1) else 0) - 1
    connections = np.vstack((connections, lines[point_index]))
    connections = np.vstack((connections,
np.matmul(connections[-1],mirror_matrix)))
    if depth == 1:
```

```

        connections = connections[2:]

        return get_objects(triangles, lines, connections, depth + 1)
    else:
        return basic_triangles, np.append(basic_lines, connections, axis=0)

def draw(basic_triangles, basic_lines):
    triangles = basic_triangles
    lines = basic_lines
    if order-1:
        triangles, lines = get_objects(triangles, basic_lines,
np.array([[0,0],[0,0]]), 1)
        glPolygonMode(GL_FRONT_AND_BACK, GL_LINE)
        glBegin(GL_TRIANGLES)
        glColor3f(1.0, 0.0, 0.0)
        for index in range(0, triangles.shape[0], 4):
            glVertex2f(*(triangles[index]))
            glVertex2f(*(triangles[index + 1]))
            glVertex2f(*(triangles[index + 2]))

            glVertex2f(*(triangles[index + 1]))
            glVertex2f(*(triangles[index + 2]))
            glVertex2f(*(triangles[index + 3]))
        glEnd()
        glColor4f(255,255,255,1.0)
        glBegin(GL_LINES)
        for index in range(0, lines.shape[0], 2):
            glVertex2f(*lines[index])
            glVertex2f(*lines[index+1])
        glEnd()

def display():
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT |
GL_STENCIL_BUFFER_BIT)
        glMatrixMode(GL_MODELVIEW)
        glLoadIdentity()

```

```
basic_triangles = np.array([[ -0.96, -0.48], [0, -0.48], [0, 0.48],  
[0.96, -0.48]])  
basic_lines = np.array([[ -0.32, -0.16], [0.32, -0.16]])  
draw(basic_triangles, basic_lines)
```

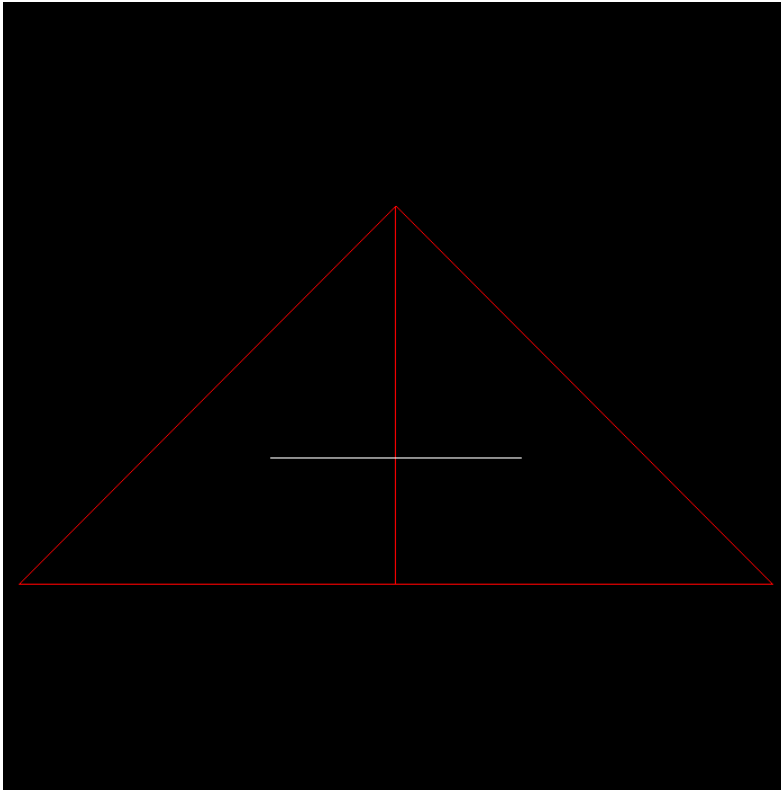
Процесс построения фрактала:

1. Изначальные объекты: равнобедренный прямоугольный треугольник и прямая, перпендикулярная к высоте треугольника, отстоящая от основания  $\frac{1}{3}$  длины высоты.
2. Пока текущий уровень не равен необходимому объектам предыдущего уровня копируются:
  - a. уменьшается в  $\sqrt{2}$  раз
  - b. поворачиваются на угол  $\frac{3*\pi}{4}$
  - c. сдвигаются влево вниз на восьмую часть длины основания
  - d. проводится повторное действие, но угол  $\frac{-3*\pi}{4}$ , сдвиг вправо вниз на то же значение.
  - e. перейти к шагу 2.

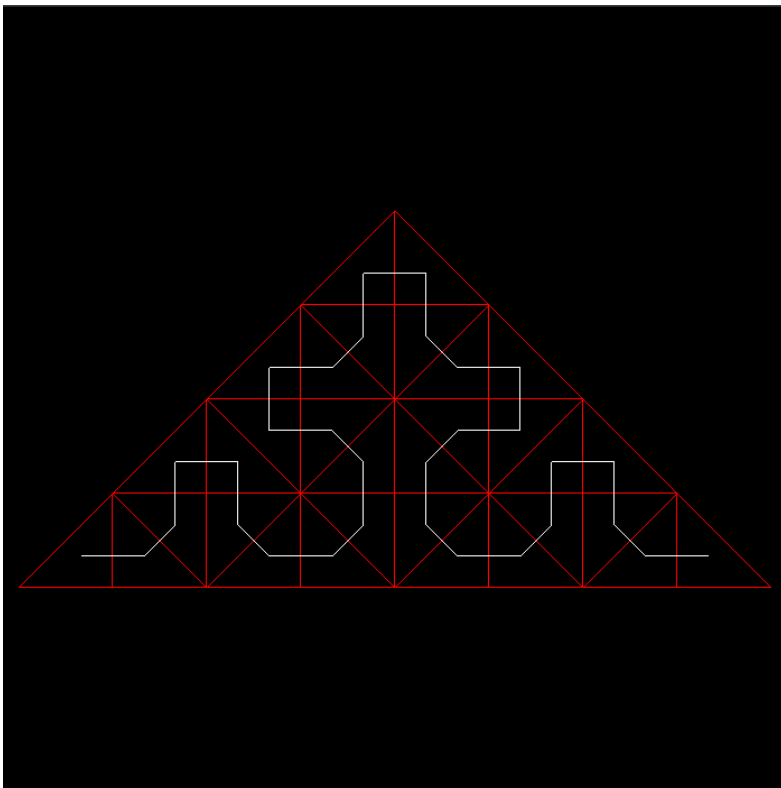
## Тестирование

Результаты тестирования представлены на снимках экрана.

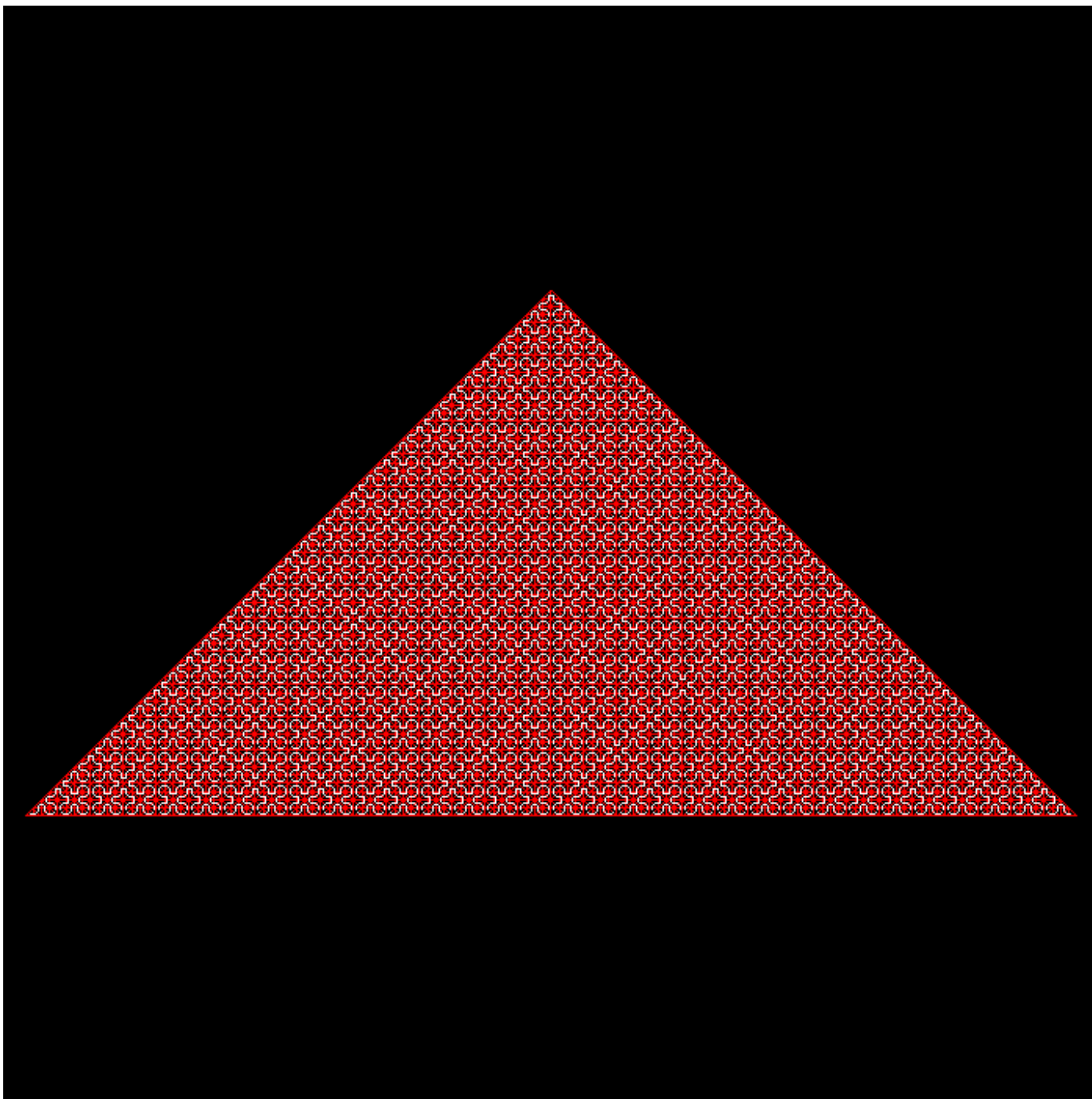
1. Первый уровень



## 2. Пятый уровень



### 3. 13 уровень



**Вывод.** В процессе выполнения лабораторной работы была разработана программа, отображающая фрактал. При



выполнении работы были приобретены  
навыки работы с графической библиотекой  
OpenGL.