

УДК 004.438

С. А. Канаев,

факультет педагогики, менеджмента и информационных технологий в образовании,

Филиал Омского государственного педагогического университета в г. Таре

Научный руководитель: ст. преподаватель О. И. Козун

Черепашня графика для создания фракталов

Аннотация. В статье рассматривается применение черепашьей графики к построению фракталов из простых графических объектов на языке программирования Python с использованием графического модуля Turtle. Также приводятся примеры кода и результаты построения.

Ключевые слова: Python, черепашня графика, модуль Turtle, программирование, фрактальная графика.

Черепашня графика — это метод программирования, который позволяет создавать простые графические объекты, в том числе и фракталы, управляя виртуальной черепашкой, которая может двигаться вперед, назад, поворачиваться вправо или влево и оставлять или не оставлять за собой след — линию. Модуль Turtle и язык программирования Python предоставляют обширный набор команд для управления черепашкой [2].

Фракталом (от лат. fractus — дробленный, сложенный) называется множество, обладающее свойством самоподобия, когда увеличенные части объекта схожи с ним самим и друг с другом. Основными достоинствами фрактала являются возможность создания сложных композиций,

наглядность и безграничная масштабируемость. Фрактальная графика позволяет получать интересные и особенные изображения, используя один и тот же алгоритм и изменяя лишь некоторые его параметры. При помощи фракталов можно представить некоторые природные объекты: деревья, облака, снежинки [1].

С помощью алгоритмов черепашьей графики можно запрограммировать фрактальные кривые. Текущее положение черепашки можно задать координатами по горизонтали и вертикали и углом, определяющим направление движения. Длина шага черепашки и изменение угла поворота существенно влияют на получаемое изображение графических объектов [3].

```

1 import turtle
2 # Создаем объект черепашки
3 my_turtle = turtle.Turtle()
4 my_turtle.speed(0) # Максимальная скорость
5 my_turtle.hideturtle()
6 my_turtle.color("darkgreen")
7 def koch(turtle, length, depth):
8     """Рекурсивная функция для рисования кривой Коха."""
9     if depth == 0:
10         turtle.forward(length)
11     else:
12         koch(turtle, length / 3, depth - 1)
13         turtle.left(60)
14         koch(turtle, length / 3, depth - 1)
15         turtle.right(120)
16         koch(turtle, length / 3, depth - 1)
17         turtle.left(60)
18         koch(turtle, length / 3, depth - 1)
19 def koch_snowflake(turtle, length, depth):
20     """Рисуем снежинку Коха."""
21     for i in range(3):
22         koch(turtle, length, depth)
23         turtle.right(120)
24 # Настраиваем параметры
25 length = 200
26 depth = 3 # Глубина рекурсии (чем больше, тем сложнее и красивее)
27 # Перемещаем черепашку в начальную позицию,
28 # чтобы снежинка была в центре
29 my_turtle.penup()
30 my_turtle.goto(-length / 2, length / 3) # Center
31 my_turtle.pendown()
32 # Рисуем снежинку Коха
33 koch_snowflake(my_turtle, length, depth)
34 turtle.done()

```

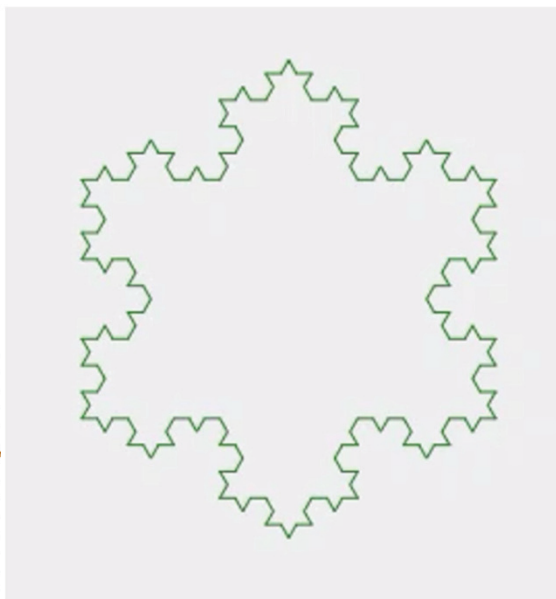


Рис. 1. Пример кода и построение фрактала «Снежинка Коха»

```

1 import turtle
2 # Создание черепахи
3 my_turtle = turtle.Turtle()
4 my_turtle.speed(0)
5 my_turtle.hideturtle()
6 my_turtle.color("blue")
7 # Функция для рисования треугольника
8 def draw_triangle(turtle, length):
9     for _ in range(3):
10         turtle.forward(length)
11         turtle.left(120)
12 # Функция для рисования треугольника Серпинского
13 def sierpinski(turtle, length, depth):
14     if depth == 0:
15         draw_triangle(turtle, length)
16     else:
17         sierpinski(turtle, length / 2, depth - 1)
18         turtle.forward(length / 2)
19         sierpinski(turtle, length / 2, depth - 1)
20         turtle.backward(length / 2)
21         turtle.left(60)
22         turtle.forward(length / 2)
23         turtle.right(60)
24         sierpinski(turtle, length / 2, depth - 1)
25         turtle.left(60)
26         turtle.backward(length / 2)
27         turtle.right(60)
28 # Начальная позиция
29 my_turtle.penup()
30 my_turtle.goto(-200, -150) # Центрирование треугольника
31 my_turtle.pendown()
32 # Рисование треугольника Серпинского
33 sierpinski(my_turtle, 400, 4) # Длина = 400, глубина = 4
34 # Завершение работы
35 turtle.done()

```

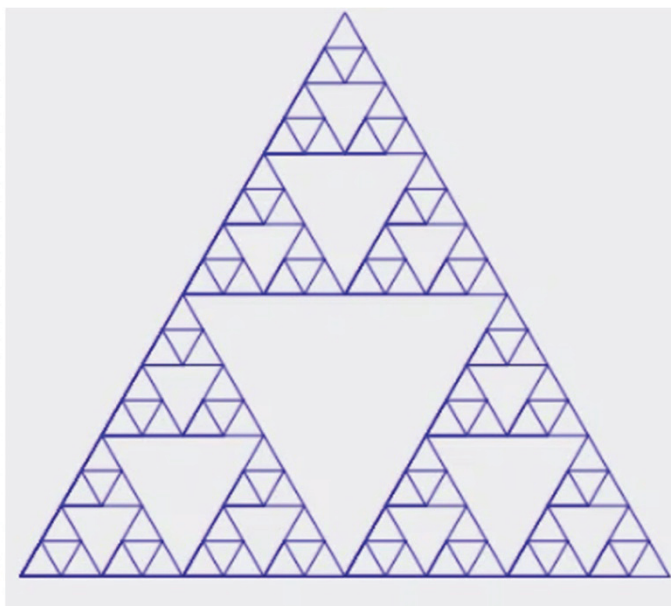


Рис. 2. Пример кода и построение треугольника Серпинского

Реализуем в Python с использованием модуля Turtle наиболее известные фракталы: снежинку Коха и треугольник Серпинского.

Для рисования первого объекта «Снежинка Коха» (рис. 1) используем рекурсивную функцию глубиной 3, которая повторяет процесс построения кривой: промежуток разделяется на три равных сегмента, средний из которых замещается треугольником. Процесс повторяется для каждого сегмента. Глубина вложения рекурсии увеличивает сложность фрактала. Рисование фигуры осуществляется при помощи команд черепащей графики.

Для реализации фрактала «Треугольник Серпинского» (рис. 2) также используем рекурсивную функцию глубины 4, которая делит равносторонний треугольник на меньшие подобные фигуры. Затем при помощи команд черепащей графики происходит рисование фрактала.

Большинство известных фрактальных кривых можно запрограммировать. Создание программ для рисования фракталов с использованием стандартного модуля Turtle не требует обширных знаний языка программирования Python.

1. Жуков А. С., Бизюк В. В. Фрактальная графика с использованием L-систем и языка Python // Студенческая наука Подмосквю : сб. материалов Междунар. науч. конф. молодых ученых. — Орехово-Зуево : Гос. гуманитар.-технол. ун-т, 2022. — С. 247–250.

2. Лужков А. А., Тюканов А. С. Основы работы в Python : учеб.-метод. пособие. — СПб. : Изд-во Рос. гос. пед. ун-та им. А. И. Герцена, 2022. — 76 с.

3. Мамыкин А. Е., Корнилаева А. А. Построение фрактальных объектов на основе L-систем // Мавлютовские чтения : материалы XVI Всерос. молодеж. науч. конф. : в 6 т. — Уфа : Уфим. гос. авиац. техн. ун-т, 2022. — Т. 5. — С. 1048–1053.