Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №7**

**«Исследование фракталов»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-101-51-00

Пешнин Владислав Константинович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

В отчете должны отображаться:

1. Цель работы

Цель работы. получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.

1. Скриншот задания (с вариантом)

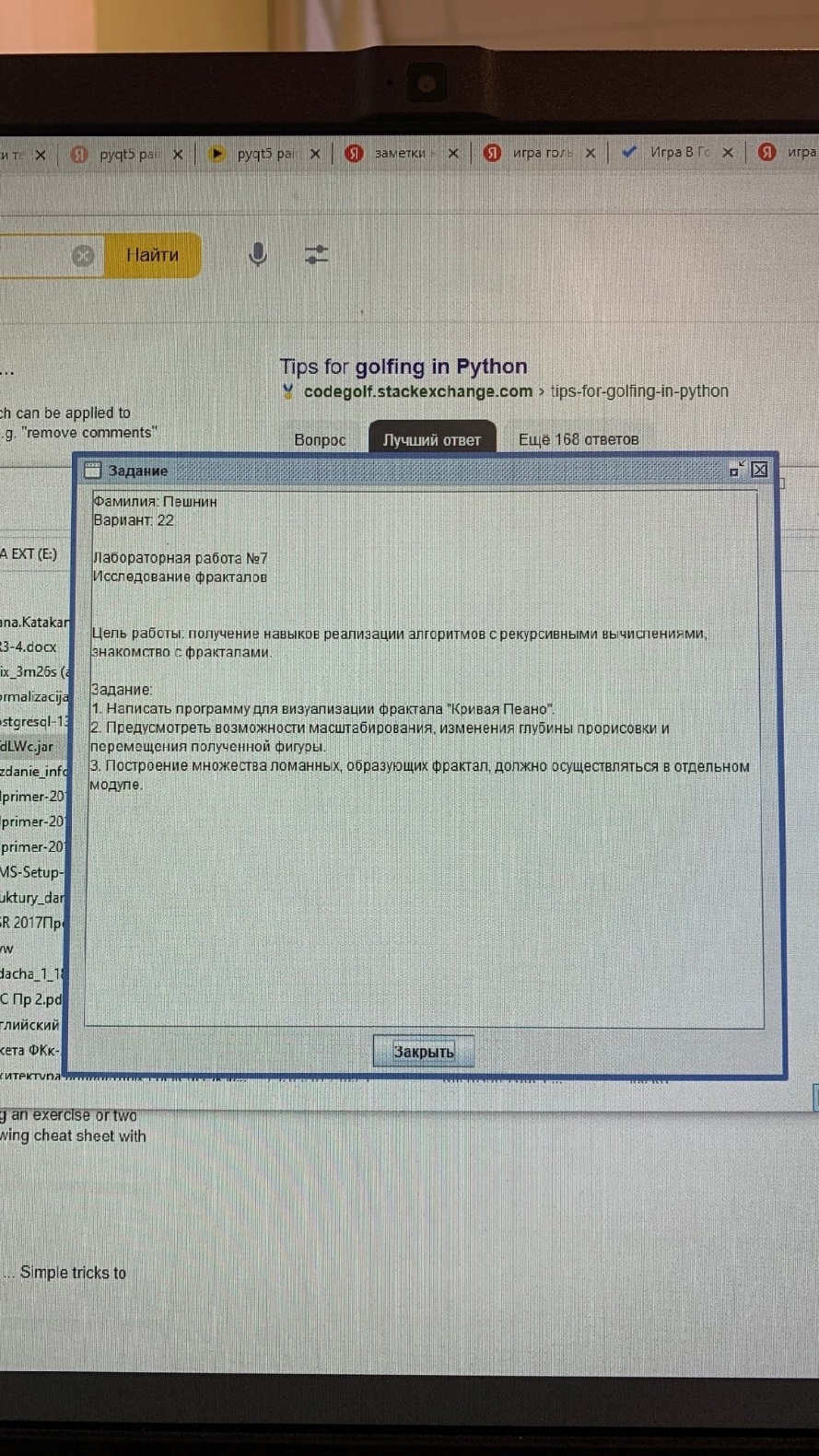


Рисунок 1- скриншот задания

1. Описание алгоритма

Для начала оформляем процесс взаимодействия с ide, через условия. При увеличении а = 2, при уменьшении а = 0.5. Для взаимодействия с самим фракталом используем x и y координаты. Следующим действием настраиваем само перо через def CurvesTurtle(), потом указываем действия с фракталом, def main() это процесс рисования и набора условия для рисования.

1. Код программы

from turtle import \*  
import turtle  
  
a = 1  
p=1  
z=1  
  
print('Увеличить',  
 'Уменьшить',  
 'Выход',  
 'Налево',  
 'Направо',  
 'Вверх',  
 'Вниз')  
  
x = input()  
for i in x:  
 if x == "Увеличить":  
 a = 2  
 turtle.setup(1000, 1000)  
 elif x == "Уменьшить":  
 a = 0.5  
 turtle.setup(1000, 1000)  
 if x == "Выход":  
 quit()  
 if x == "Направо":  
 p = 150  
 elif x == "Вверх":  
 z = 150  
 elif x == "Вниз":  
 z = -150  
 elif x == "Налево":  
 p = - 150  
  
  
  
  
class CurvesTurtle(Pen):  
 def hilbert(self, size, level, parity):  
 if level == 0:  
 return  
 self.left(parity \* 90)  
 self.hilbert(size, level - 1, -parity)  
 self.forward(size)  
 self.right(parity \* 90)  
 self.hilbert(size, level - 1, parity)  
 self.forward(size)  
 self.hilbert(size, level - 1, parity)  
 self.right(parity \* 90)  
 self.forward(size)  
 self.hilbert(size, level - 1, -parity)  
 self.left(parity \* 90)  
  
 def fractal(self, dist, depth, dir):  
 if depth < 1:  
 self.fd(dist)  
 return  
 self.fractal(dist / 3, depth - 1, dir)  
 self.lt(60 \* dir)  
 self.fractal(dist / 3, depth - 1, dir)  
 self.rt(120 \* dir)  
 self.fractal(dist / 3, depth - 1, dir)  
 self.lt(60 \* dir)  
 self.fractal(dist / 3, depth - 1, dir)  
  
  
def main():  
 ft = CurvesTurtle()  
 ft.reset()  
 ft.speed(0)  
 ft.ht()  
 ft.getscreen().tracer(1, 0)  
 ft.pu()  
 size = 6 \* a  
 ft.setpos(-33 \* size + p, -32 \* size + z)  
 ft.pd()  
 ft.fillcolor("red")  
 ft.begin\_fill()  
 ft.fd(size)  
 ft.hilbert(size, 6, 1)  
 ft.fd(size)  
 for i in range(3):  
 ft.lt(90)  
 ft.fd(size \* (64 + i % 2))  
 ft.pu()  
 for i in range(2):  
 ft.fd(size)  
 ft.rt(90)  
 ft.pd()  
 for i in range(4):  
 ft.fd(size \* (66 + i % 2))  
 ft.rt(90)  
 ft.end\_fill()  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 msg = main()  
 print(msg)  
 mainloop()

1. Результат выполнения программы

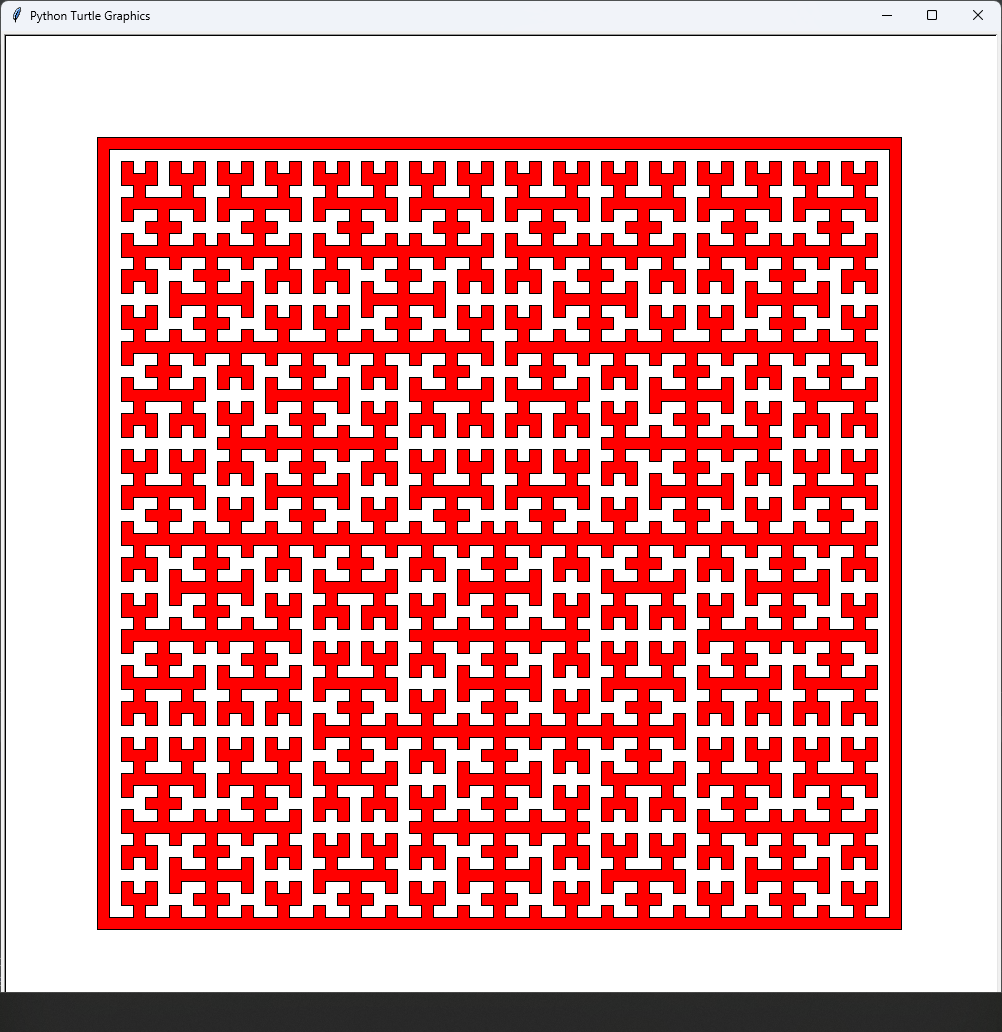


Рисунок – 2 увеличение фрактала

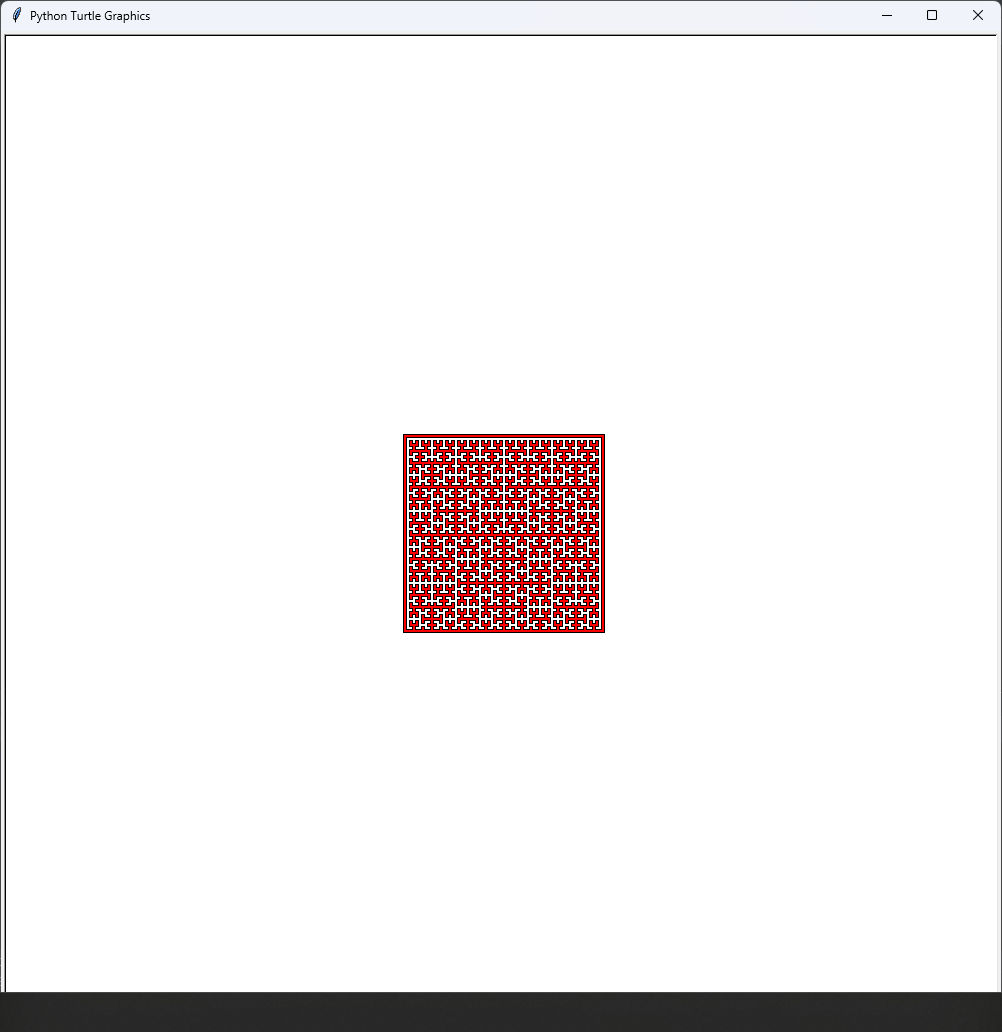


Рисунок 3 – уменьшение фрактала

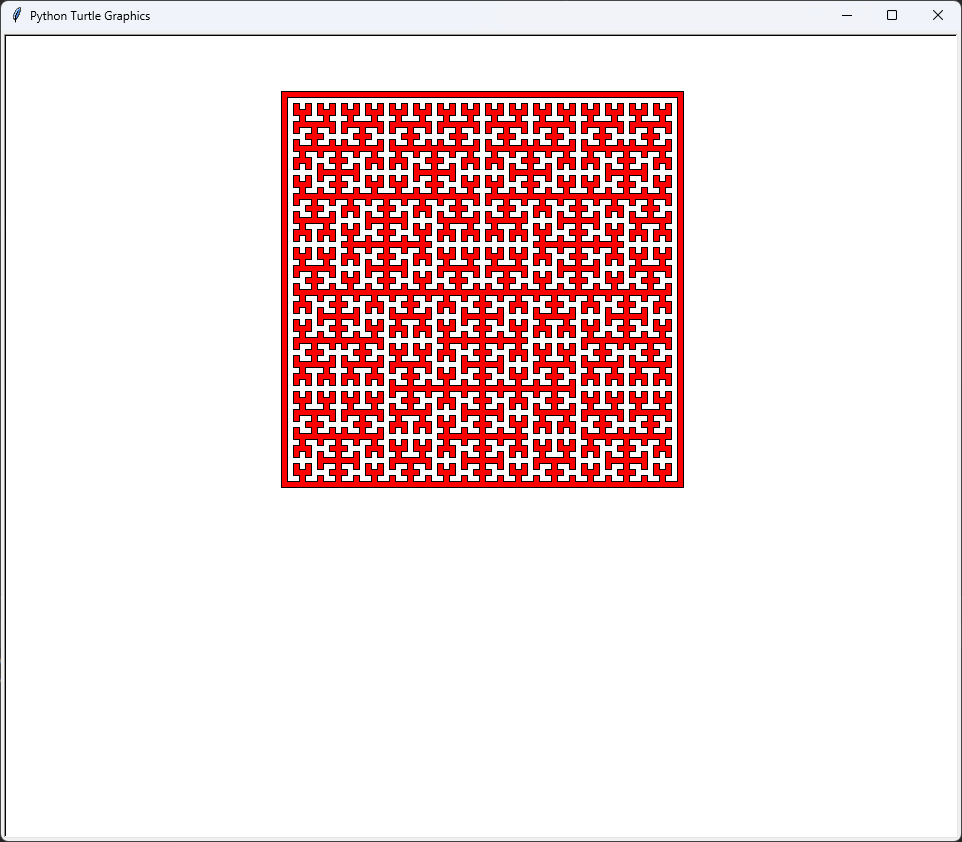


Рисунок 4 - перемещение фрактала вверх

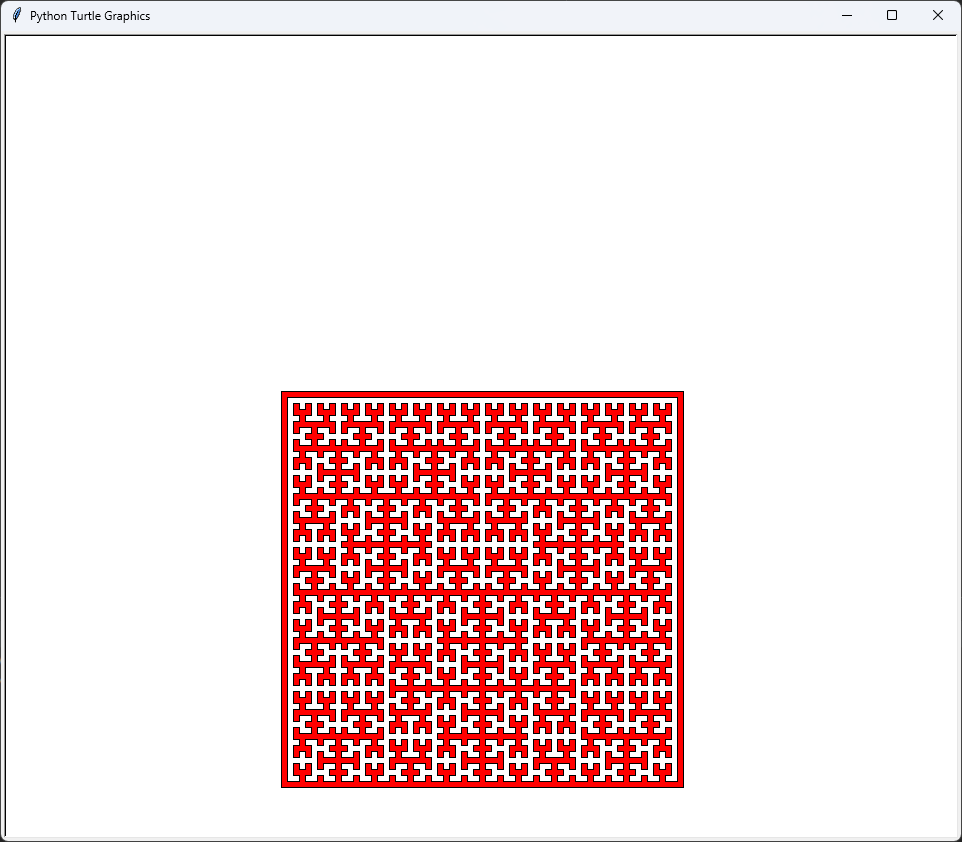


Рисунок 5 – перемещение фрактала вниз

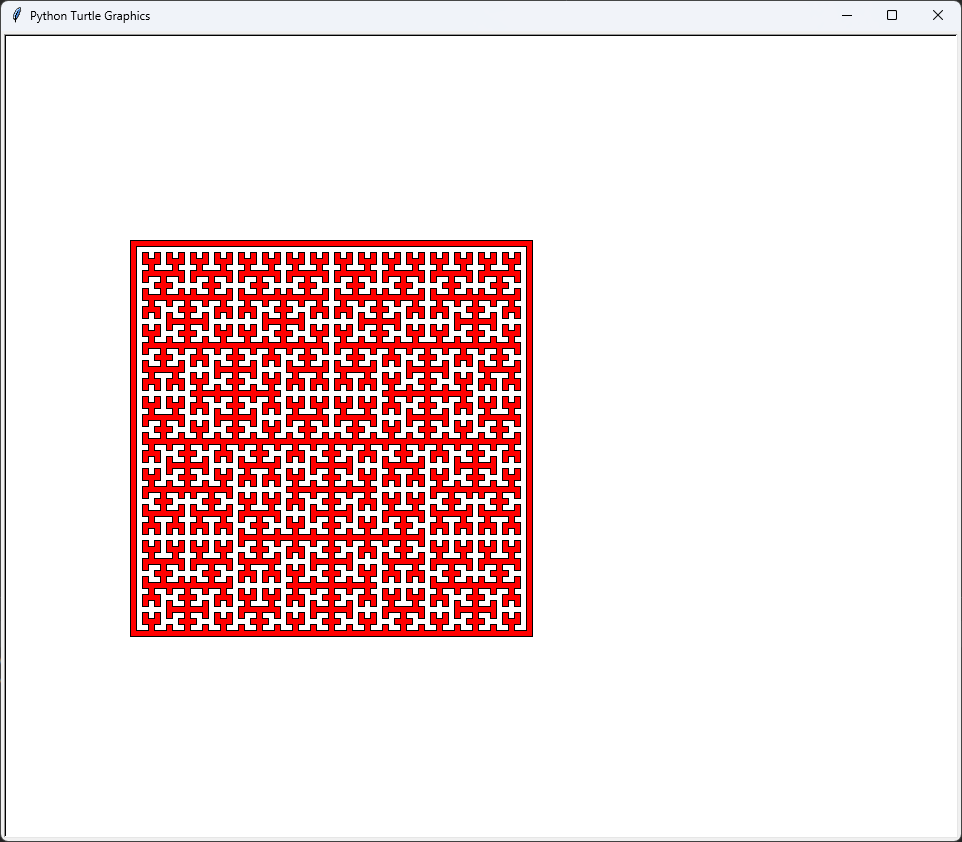


Рисунок 6 – перемещение фрактала влево

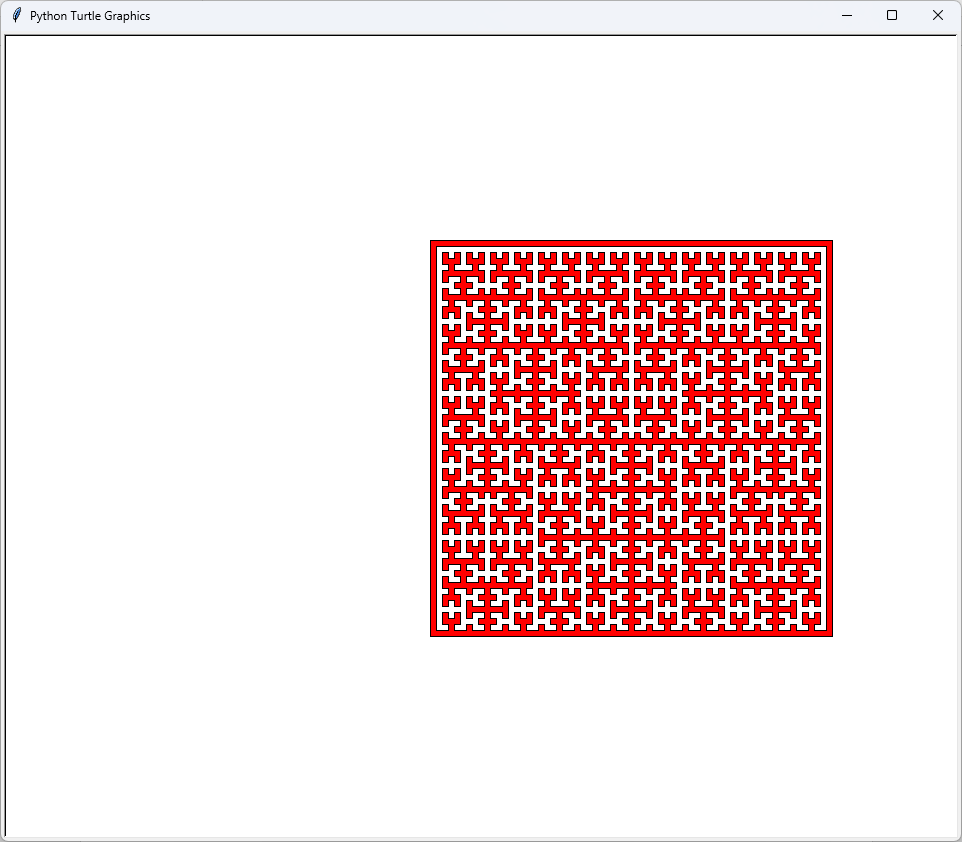


Рисунок 7 – перемещение фрактала вправо

1. Вывод

В ходе работы получили навыки реализации алгоритмов с рекурсивными вычислениями. Познакомились с фракталами. Узнали новые алгоритмы вычисления и заполнения фракталов с помощью новых изученных функций.

Проблемы, с которыми столкнулся:

Не достаточность знаний с использованными функциями и реализации фрактала.