Министерство цифрового развития, связи

и массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Лабораторная работа №5

по дисциплине «Структуры и алгоритмы обработки данных»

“Рекурсия. Фракталы.”

Выполнила:

студентка группы БСТ2004

Моисеева С.Д.

Проверил:

А.Д. Чайка

Москва 2022

**Цель работы**: изучение основных алгоритмов рекурсии и фракталов и их реализация на Python

**Задание:** Реализовать генерацию фрактала ковра Серпинского с применением рекурсивных функций.

Добавить возможность задания глубины фрактала.

Оценить глубину рекурсии.

Построить таблицу зависимости времени построения от глубины фрактала.

**Ход работы.**

**Задание 1**

Ниже представлена реализация ковра Серпинского с использованием вложенных массивов:

import time  
  
  
class Rug:  
 def \_\_init\_\_(self, depth):  
 self.pattern = [[0, 0, 0], [0, 1, 0], [0, 0, 0]]  
 self.pattern1 = [[1, 1, 1], [1, 1, 1], [1, 1, 1]]  
 self.rug = [[0, 0, 0], [0, 1, 0], [0, 0, 0]]  
 self.depth = depth  
 if self.depth < 2:  
 self.printable\_rug = [[0, 0, 0], [0, 1, 0], [0, 0, 0]]  
 else:  
 self.printable\_rug = [[[] for \_ in range(3 \*\* (self.depth - 1))] for \_ in range(3 \*\* self.depth)]  
 self.go\_deeper(self.rug, 1)  
  
 def go\_deeper(self, piece, curr\_depth):  
 if curr\_depth < self.depth:  
 for i in range(3):  
 for j in range(3):  
 if piece[i][j] != 1:  
 piece[i][j] = self.pattern  
 # для демонстрации на примере с глубиной 2  
 self.printable\_rug[i \* 3][j] = self.pattern[0]  
 self.printable\_rug[i \* 3 + 1][j] = self.pattern[1]  
 self.printable\_rug[i \* 3 + 2][j] = self.pattern[2]  
 else:  
 piece[i][j] = self.pattern1  
 # для демонстрации на примере с глубиной 2  
 self.printable\_rug[i \* 3][j] = self.pattern1[0]  
 self.printable\_rug[i \* 3 + 1][j] = self.pattern1[1]  
 self.printable\_rug[i \* 3 + 2][j] = self.pattern1[2]  
 self.go\_deeper(piece[i][j], curr\_depth + 1)  
 return 0  
  
  
depth = int(input('Введите желаемую глубину ковра :)'))  
t = time.perf\_counter()  
the\_rug = Rug(depth)  
res = 'Время выполнения на глубине ' + str(the\_rug.depth) + ': {0:.6f} сек'.format((time.perf\_counter() - t))  
print(res)  
with open('matrixres.txt', 'a') as file:  
 file.write(res + "\n")  
print('Глубина рекурсии:', the\_rug.depth)  
print("Представление многомерным массивом (с вложениями):")  
print(the\_rug.rug)  
print("Представление матрицей:")  
for i in the\_rug.printable\_rug:  
 print(i)

Результат выполнения задания представлен на рисунке 1:

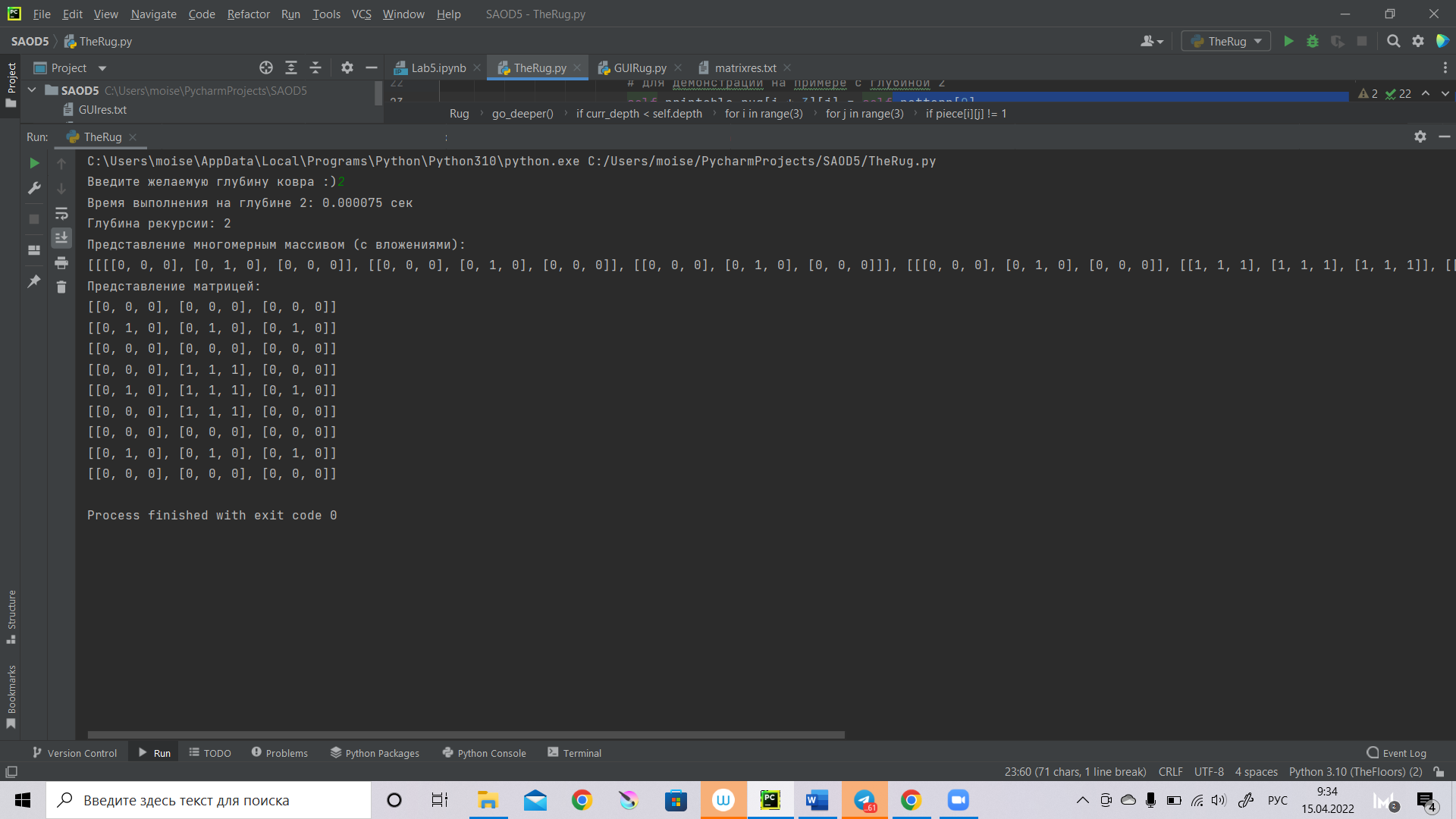


Рисунок 1 – Тестирование консольной реализации

Ниже представлена реализация того же фрактала с использованием библиотеки turtle. Отличие от первого алгоритма заключается в том, что в данном случае фрактал рисуется с максимальной глубины.

import random  
import time  
import turtle  
  
  
def s(n, m, color):  
 global colors  
 if n == 0:  
 turtle.begin\_fill()  
 for \_ in range(4):  
 turtle.color('black')  
 turtle.forward(m)  
 turtle.left(90)  
 turtle.color(colors[random.randrange(0, len(colors))])  
 turtle.end\_fill()  
  
 else:  
 for \_ in range(4):  
 s(n - 1, m / 3, (color + 1) % len(colors))  
 turtle.color('black')  
 turtle.forward(m / 3)  
 s(n - 1, m / 3, (color + 1) % len(colors))  
 turtle.color('black')  
 turtle.forward(m / 3)  
 turtle.color('black')  
 turtle.forward(m / 3)  
 turtle.left(90)  
  
  
colors = ['red', 'orange', 'yellow', 'green', 'cyan', 'blue', 'magenta']  
depth = int(input('Введите желаемую глубину ковра :)'))  
turtle.tracer(10)  
turtle.up()  
turtle.setpos(-200, -200)  
turtle.down()  
turtle.speed(100)  
t = time.perf\_counter()  
s(depth, 400, 0)  
res = 'Время выполнения на глубине ' + str(depth) + ': {0:.6f} сек'.format((time.perf\_counter() - t))  
print(res)  
with open('GUIres.txt', 'a') as file:  
 file.write(res + "\n")  
turtle.done()

Результат работы программы представлен на рисунке 2.

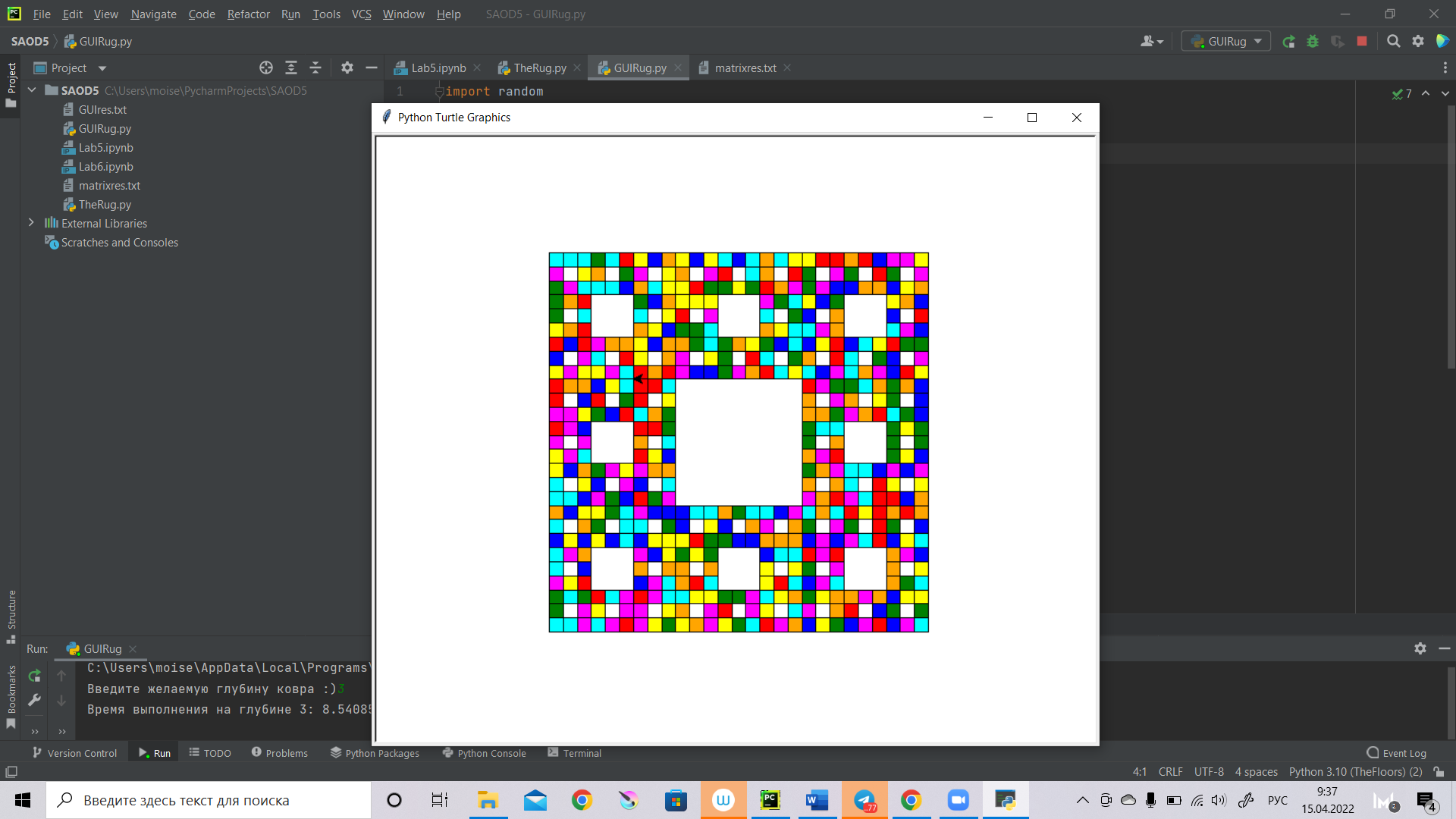


Рисунок 2 – Графическое решение на turtle

Вывод: Фрактал был реализован на основе рекурсии

Ссылка на github: https://github.com/mmaider/SAOD5

**Список литературы**

1. Камаев В.А., Костерин В.В. Технологии программирования. М.: Высшая школа,

2006.

2. Жоголев Е.А.Технология программирования. – М.: Научный мир, 2004.

Git:

3. Scott Chacon, Ben Straub «Pro Git»

4. git-scm.com