# 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-204-52-00

Федяев Никита Юрьевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**1. Цель работы:** получение навыков организации алгоритмов с рекурсивными вычислениями. Знакомство с фракталами

**2. Формулировка задания**

1. Написать программу на паскале для визуализации фрактала "Ковер Серпинского". Предусмотреть возможность масштабирования, изменения глубины прорисовки и перемещение полученной фигуры. Построение множества ломанных должно осуществляться в отдельном модуле.

**3. Описание алгоритма**

Ковер Серпинского – это фрактальная структура, которая состоит из квадратов, которые повторяются и заполняют плоскость. Каждый квадрат делится на девять более маленьких квадратов и серединный квадрат вырезается

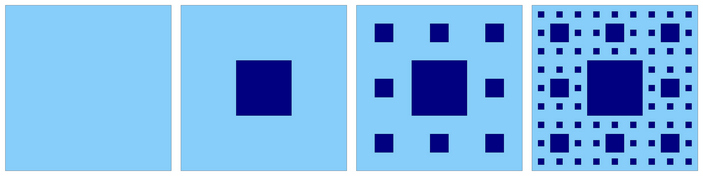


Рис.1. ломанная-генератор глубина 0,1,2 и 3

**4. Схема алгоритма с комментариями**

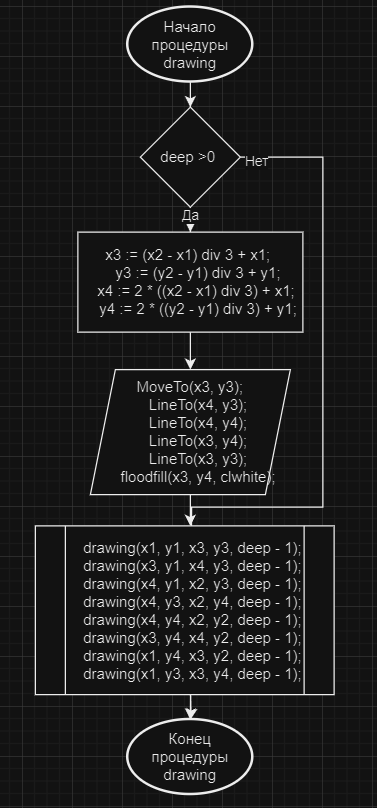


Рис 2. Структура процедуры drawing

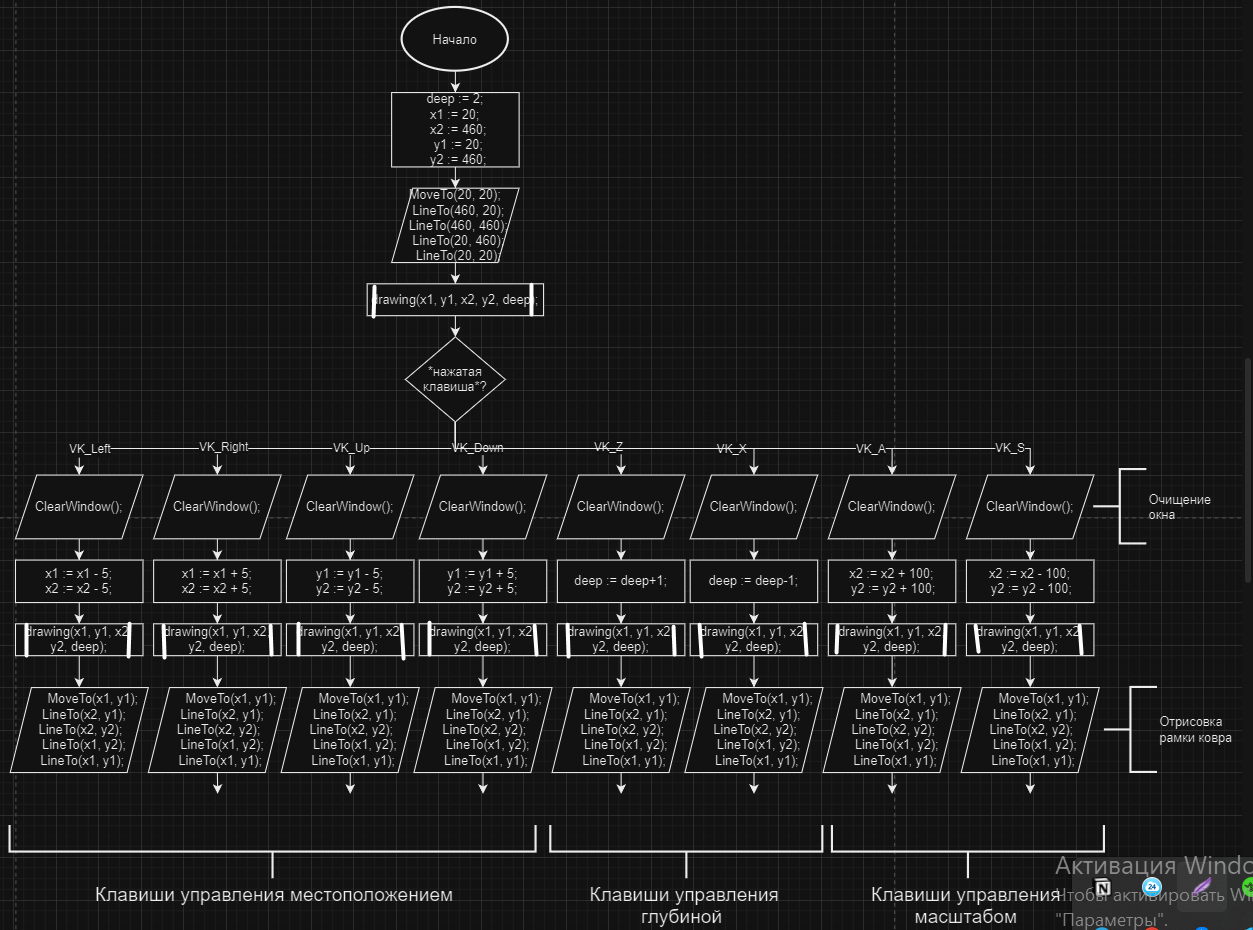


Рис 3. Код main.pas

**5. Код программы**

**main.pas**

**uses**

GraphABC, zxc;

**procedure** KeyPressed(Key: integer);

**begin**

**case** Key **of**

VK\_Left:

**begin**

ClearWindow();

x1 := x1 - 5;

x2 := x2 - 5;

drawing(x1, y1, x2, y2, deep);

MoveTo(x1, y1);

LineTo(x2, y1);

LineTo(x2, y2);

LineTo(x1, y2);

LineTo(x1, y1);

**end**;

VK\_Right:

**begin**

ClearWindow();

x1 := x1 + 5;

x2 := x2 + 5;

drawing(x1, y1, x2, y2, deep);

MoveTo(x1, y1);

LineTo(x2, y1);

LineTo(x2, y2);

LineTo(x1, y2);

LineTo(x1, y1);

**end**;

VK\_Up:

**begin**

ClearWindow();

y1 := y1 - 5;

y2 := y2 - 5;

drawing(x1, y1, x2, y2, deep);

MoveTo(x1, y1);

LineTo(x2, y1);

LineTo(x2, y2);

LineTo(x1, y2);

LineTo(x1, y1);

**end**;

VK\_Down:

**begin**

ClearWindow();

y1 := y1 + 5;

y2 := y2 + 5;

drawing(x1, y1, x2, y2, deep);

MoveTo(x1, y1);

LineTo(x2, y1);

LineTo(x2, y2);

LineTo(x1, y2);

LineTo(x1, y1);

**end**;

VK\_Z:

**begin**

ClearWindow();

deep := deep+1;

drawing(x1, y1, x2, y2, deep);

MoveTo(x1, y1);

LineTo(x2, y1);

LineTo(x2, y2);

LineTo(x1, y2);

LineTo(x1, y1);

**end**;

VK\_X:

**begin**

ClearWindow();

deep := deep-1;

drawing(x1, y1, x2, y2, deep);

MoveTo(x1, y1);

LineTo(x2, y1);

LineTo(x2, y2);

LineTo(x1, y2);

LineTo(x1, y1);

**end**;

VK\_A:

**begin**

ClearWindow();

x2 := x2 + 100;

y2 := y2 + 100;

drawing(x1, y1, x2, y2, deep);

MoveTo(x1, y1);

LineTo(x2, y1);

LineTo(x2, y2);

LineTo(x1, y2);

LineTo(x1, y1);

**end**;

VK\_S:

**begin**

ClearWindow();

x2 := x2 - 100;

y2 := y2 - 100;

drawing(x1, y1, x2, y2, deep);

MoveTo(x1, y1);

LineTo(x2, y1);

LineTo(x2, y2);

LineTo(x1, y2);

LineTo(x1, y1);

**end**;

**end**;

**end**;

**begin**

deep := 2;

MoveTo(20, 20);

LineTo(460, 20);

LineTo(460, 460);

LineTo(20, 460);

LineTo(20, 20);

x1 := 20;

x2 := 460;

y1 := 20;

y2 := 460;

drawing(x1, y1, x2, y2, deep);

**while** true **do**

**begin**

OnKeyDown := KeyPressed;

**end**;

**end**.

**zxc.pas**

**Unit** zxc;

**Uses** GraphABC;

**var**

x1, x2, y1, y2: integer;

deep: integer;

**procedure** drawing(x1, y1, x2, y2, deep: integer);

**var**

x3, y3, x4, y4: integer;

**begin**

**if** deep > 0 **then**

**begin**

x3 := (x2 - x1) **div** 3 + x1;

y3 := (y2 - y1) **div** 3 + y1;

x4 := 2 \* ((x2 - x1) **div** 3) + x1;

y4 := 2 \* ((y2 - y1) **div** 3) + y1;

MoveTo(x3, y3);

LineTo(x4, y3);

LineTo(x4, y4);

LineTo(x3, y4);

LineTo(x3, y3);

floodfill(x3, y4, clwhite);

setbrushcolor(clwhite);

drawing(x1, y1, x3, y3, deep - 1);

drawing(x3, y1, x4, y3, deep - 1);

drawing(x4, y1, x2, y3, deep - 1);

drawing(x4, y3, x2, y4, deep - 1);

drawing(x4, y4, x2, y2, deep - 1);

drawing(x3, y4, x4, y2, deep - 1);

drawing(x1, y4, x3, y2, deep - 1);

drawing(x1, y3, x3, y4, deep - 1);

**end**;

**end**;

**end**.

**6. Результат выполнения программы**

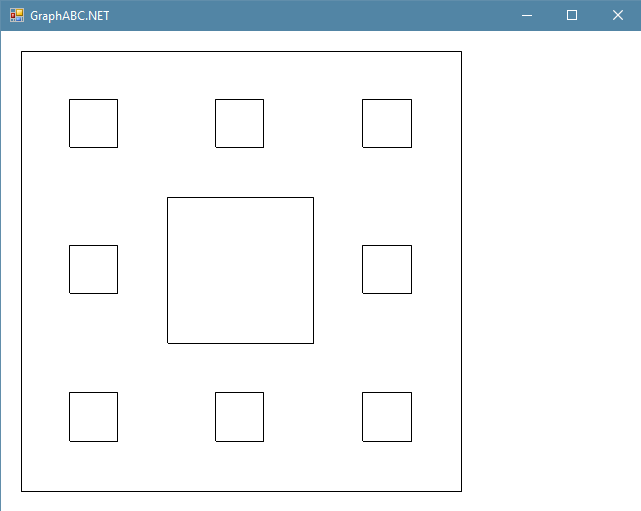


Рис 4. Ковер Серпинского.

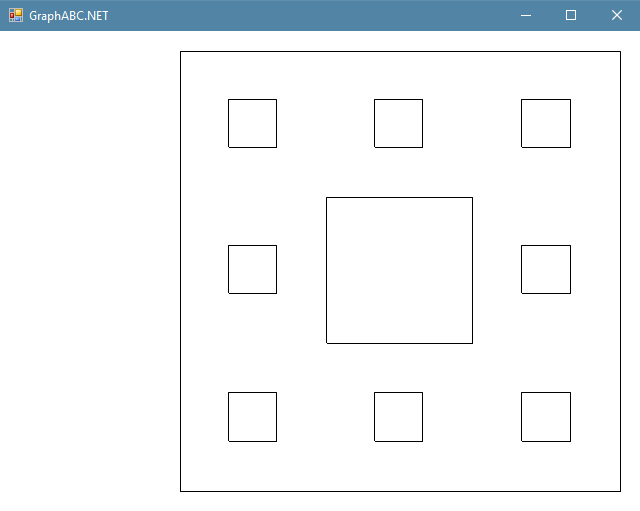


Рис 5. Ковер Серпинского движение вправо.

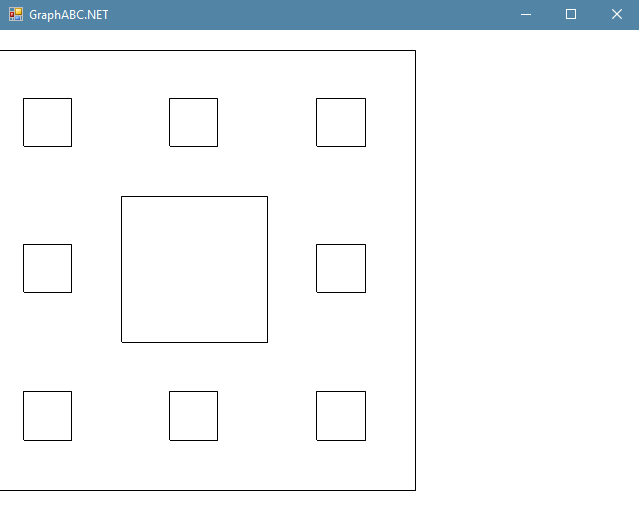


Рис 6. Ковер Серпинскогодвижение влево.

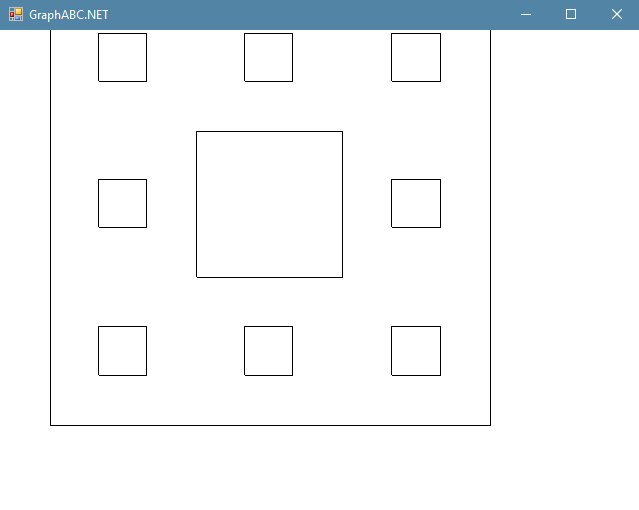


Рис 7. Ковер Серпинского движение вверх.

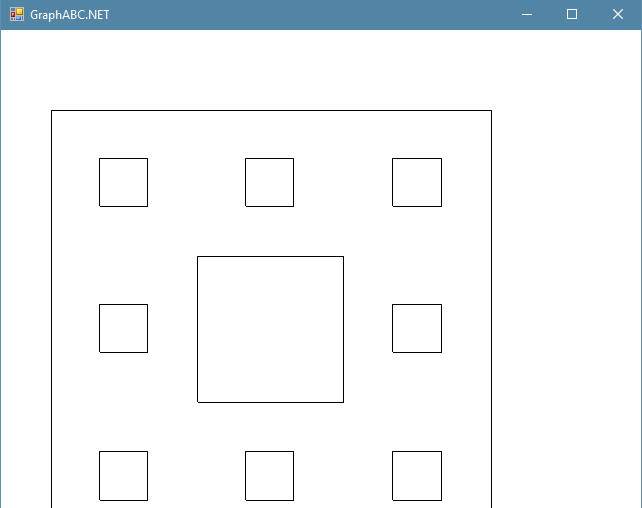


Рис 8. Ковер Серпинского движение вниз.

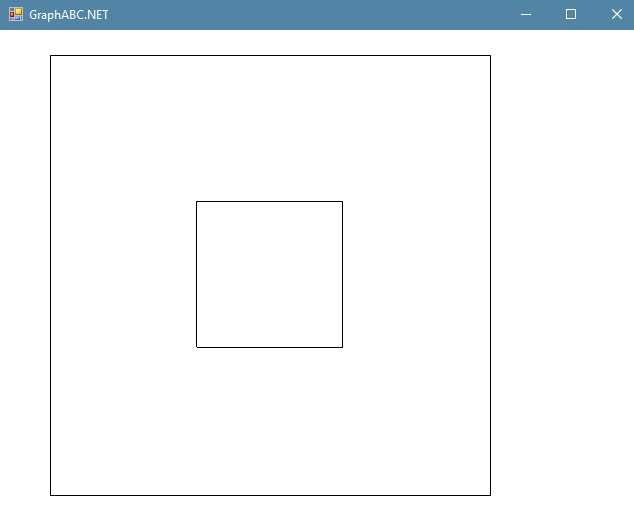


Рис 9. Ковер Серпинского уменьшение глубины.

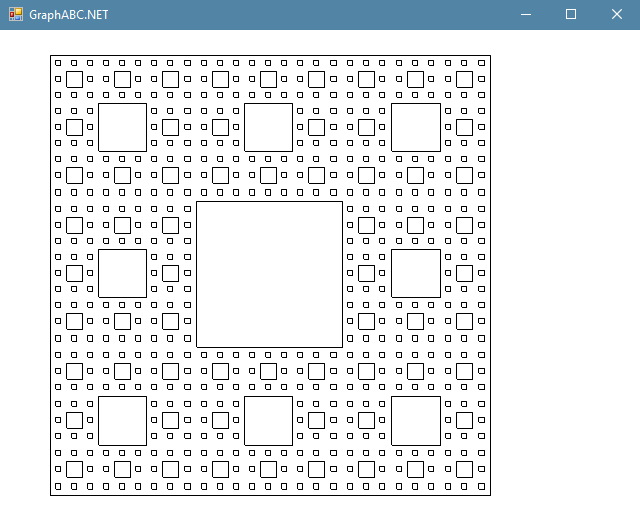


Рис 10. Ковер Серпинского увеличение глубины.

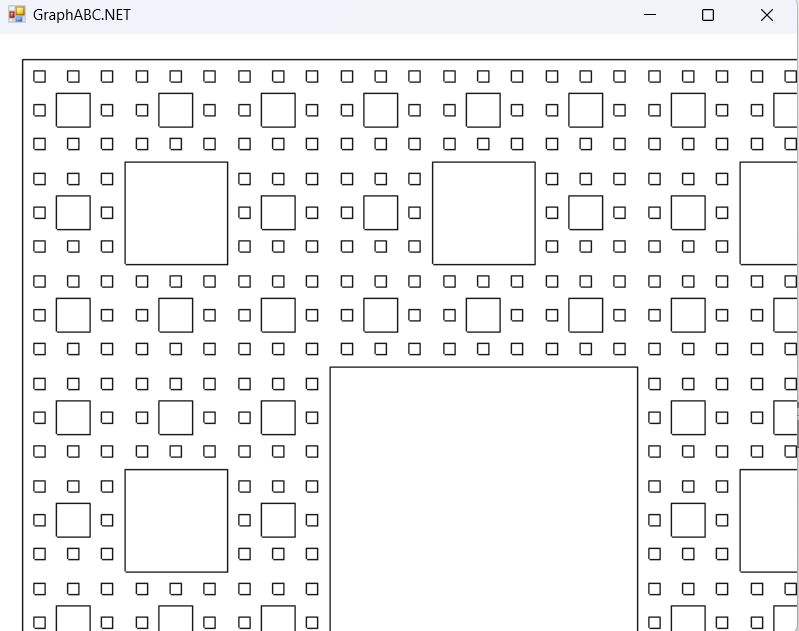


Рис 11. Ковер Серпинского увеличение масштаба.

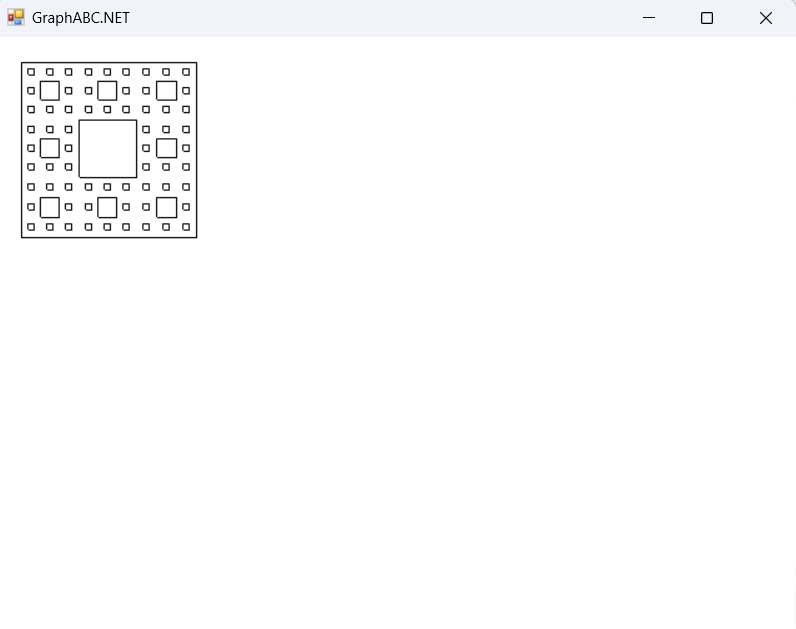


Рис 12. Ковер Серпинского уменьшение масштаба.

**7. Вывод**

В ходе работы я закрепил знания о рекурсии и написал свою рекурсивную процедуру в отдельном модуле

По ходу работы возникло много трудностей: написать рекурсивную процедуру для рисования ковра Серпинского, управлением перемещения и то как уменьшать и увеличивать глубину по нажатию клавиши. С помощью открытых источников в Интернете я нашел решения данных проблем.

В процессе выполнения работы я использовал сервис [draw.io](https://vk.com/away.php?to=http%3A%2F%2Fdraw.io&cc_key=) – это бесплатный сервис, помогающий создавать блок-схемы и диаграммы.