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»**

Выполнил: студент учебной группы

ИСПк-202-52-00

Комарова Вероника Андреевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

**1.ФОРМУЛИРОВКА ЗАДАНИЯ**

Вариант 9

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

**Постановка задачи:**

1. Написать программу для визуализации фрактала «Ковер Серпинского».
2. Предусмотреть возможность масштабирования. Изменения глубины прорисовки и перемещения полученной фигуры.
3. Построение множества ломанных, образующих фрактал, должно осуществляться в отдельном модуле.

**2. ОТВЕТЫ НА ВОПРОСЫ**

1. Фрактал — это сложная геометрическая фигура, которая может быть разделена на части, каждая из которых является уменьшенной копией всей фигуры. По сути, это самоподобие. Фракталы часто встречаются в природе, например, в структуре снежинок, деревьев, береговых линиях и облаках.
2. Начинаем с квадрата, который является исходным. Делим этот квадрат на 9 поменьше. Удаляем центральный изначальный квадрат и имеем 8 квадратов по углам. Теперь для этих 8 квадратов делаем предыдущие действия. С каждым шагом количество оставшихся квадратов будет увеличиваться, а структура будет становиться все более сложной.

**3.ОПИСАНИЕ АЛГОРИТМА**

Основной алгоритм:

1. Устанавливается размер окна
2. Устанавливается размер окна с помощью SetWindowSize(800, 800).
3. Переменные x1 и y1 инициализируются значением 0, что означает, что ковер будет нарисован с верхнего левого угла окна.
4. Переменная s инициализируется значением 800, что задает начальный размер ковра.
5. Переменная d инициализируется значением 5, что указывает на начальную глубину рекурсии.
6. Вызывается процедура Carpet
7. Запускается цикл repeat. Если change не равно 0, то начнет работать case-меню.
8. Если change равно '+', размер ковра увеличивается на 50 пикселей (s += 50), глубина увеличивается на 1 (d += 1), окно очищается с помощью ClearWindow, и вызывается Carpet(x1, y1, s, d) для перерисовки.
9. Если change равно '-', размер ковра уменьшается на 50 пикселей (s -= 50), глубина уменьшается на 1 (d -= 1), окно очищается и происходит перерисовка.
10. Если change равно 8, то координата y1 уменьшается на 50 (y1 -= 50), окно очищается и происходит перерисовка.
11. Если change равно 2, то координата y1 увеличивается на 50 (y1 += 50), окно очищается и происходит перерисовка.
12. Если change равно 4, то координата x1 уменьшается на 50 (x1 -= 50), окно очищается и происходит перерисовка.
13. Если change равно 4, то координата x1 уменьшается на 50 (x1 -= 50), окно очищается и происходит перерисовка.
14. Если change равно 6, то координата x1 увеличивается на 50 (x1 += 50), окно очищается и происходит перерисовка.
15. Внутренний цикл продолжается до тех пор, пока пользователь не введет '0'.

Алгоритм ковра Серпинского:

1. Если depth равно 0, это означает, что достигнут самый нижний уровень рекурсии. Устанавливается цвет кисти в черный (SetBrushColor(clBlack)). Рисуется квадрат с верхним левым углом в точке (x, y) и размерами size x size с помощью функции FillRect.
2. Если depth больше 0, то вычисляется новый размер квадрата: newSize := size div 3. Это делит текущий размер на три, что соответствует тому, как ковер Серпинского строится из меньших квадратов.

**4. СХЕМА АЛГОРИТМА**

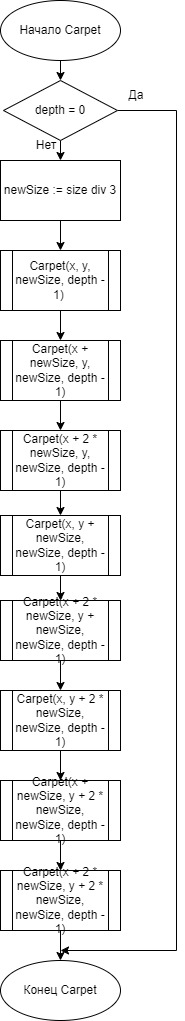


Рисунок 1 – процедура carpet

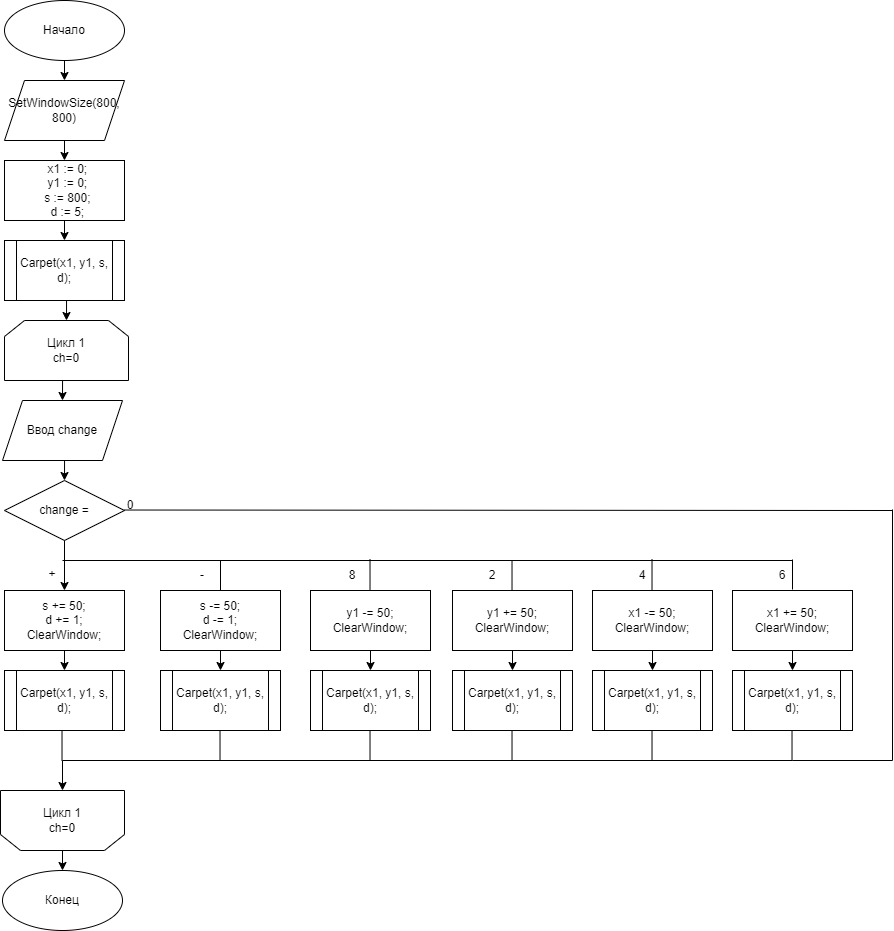


Рисунок 2 – основная часть

**5. КОД**

Код основной программы:

uses

GraphABC, m;

var

x1, y1, s, d: integer;

change: char;

begin

SetWindowSize(800, 800);

x1 := 0;

y1 := 0;

s := 800;

d := 5;

Carpet(x1, y1, s, d);

repeat

readln(change);

case change of

'+':

begin

s += 50;

d += 1;

ClearWindow;

Carpet(x1, y1, s, d);

end;

'-':

begin

s -= 50;

d -= 1;

ClearWindow;

Carpet(x1, y1, s, d);

end;

'8':

begin

y1 -= 50;

ClearWindow;

Carpet(x1, y1, s, d);

end;

'2':

begin

y1 += 50;

ClearWindow;

Carpet(x1, y1, s, d);

end;

'4':

begin

x1 -= 50;

ClearWindow;

Carpet(x1, y1, s, d);

end;

'6':

begin

x1 += 50;

ClearWindow;

Carpet(x1, y1, s, d);

end;

end;

until change = '0';

end.

Код модуля программы:

unit M;

uses

GraphABC;

procedure Carpet(x, y, size: Integer; depth: Integer);

begin

if depth = 0 then //база рекрурсии

begin

SetBrushColor(clBlack);

FillRect(x, y, x + size, y + size);

end

else

begin

var newSize := size div 3;

Carpet(x, y, newSize, depth - 1);

Carpet(x + newSize, y, newSize, depth - 1);

Carpet(x + 2 \* newSize, y, newSize, depth - 1);

Carpet(x, y + newSize, newSize, depth - 1);

Carpet(x + 2 \* newSize, y + newSize, newSize, depth - 1);

Carpet(x, y + 2 \* newSize, newSize, depth - 1);

Carpet(x + newSize, y + 2 \* newSize, newSize, depth - 1);

Carpet(x + 2 \* newSize, y + 2 \* newSize, newSize, depth - 1);

end;

end;

end.

**6. РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ**

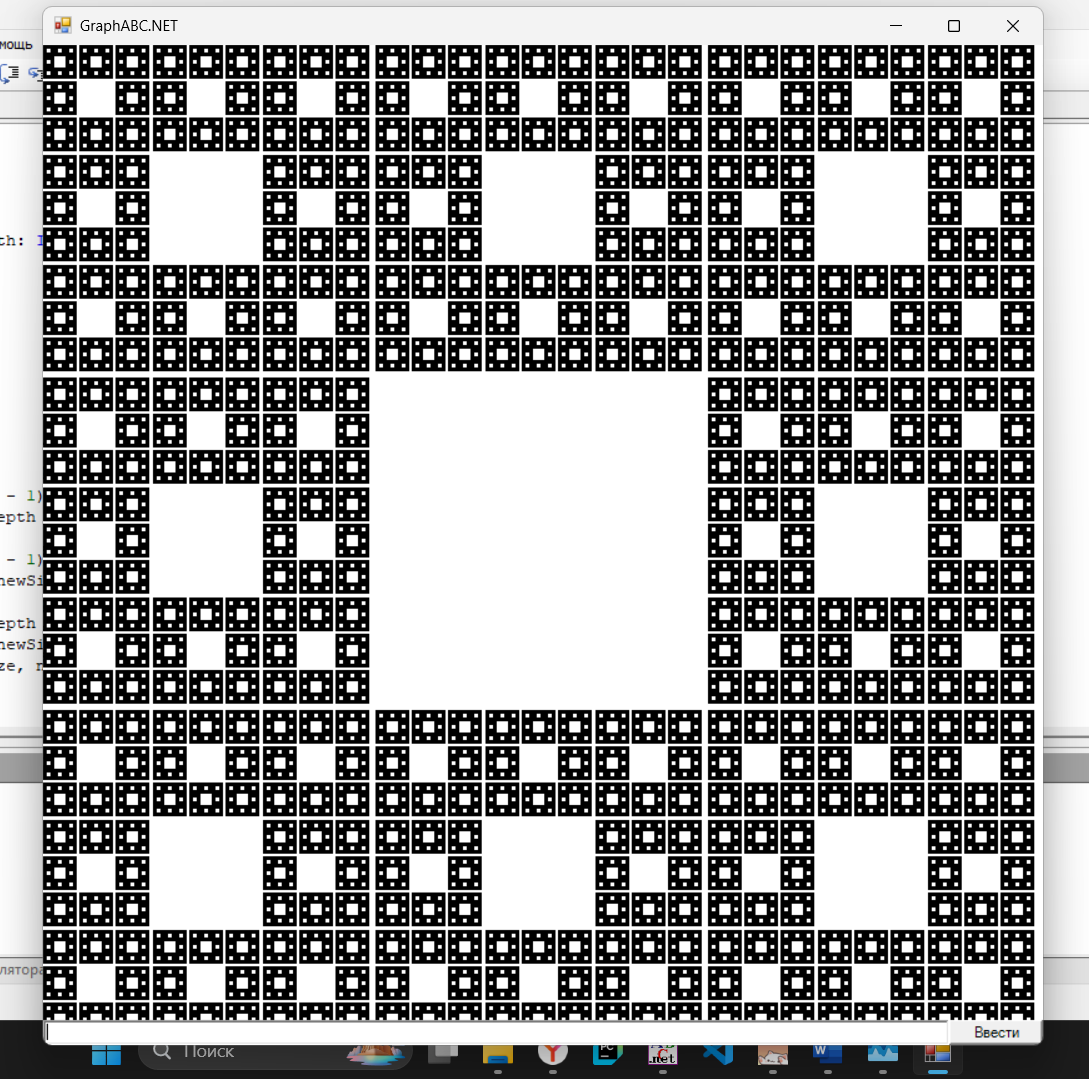


Рисунок 3 – ковер Серпинского

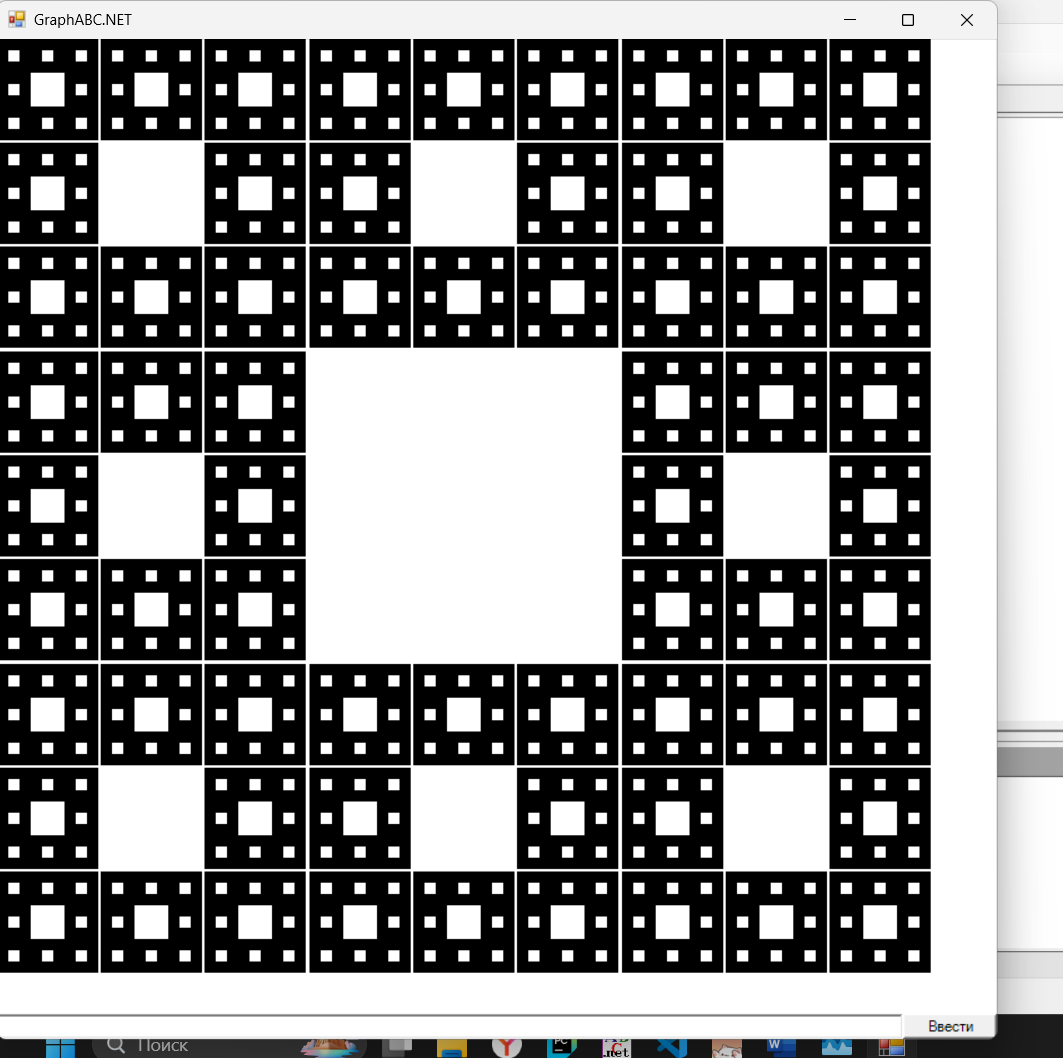


Рисунок 4 – ковер Серпинского с меньшей длиной и глубиной рекурсии

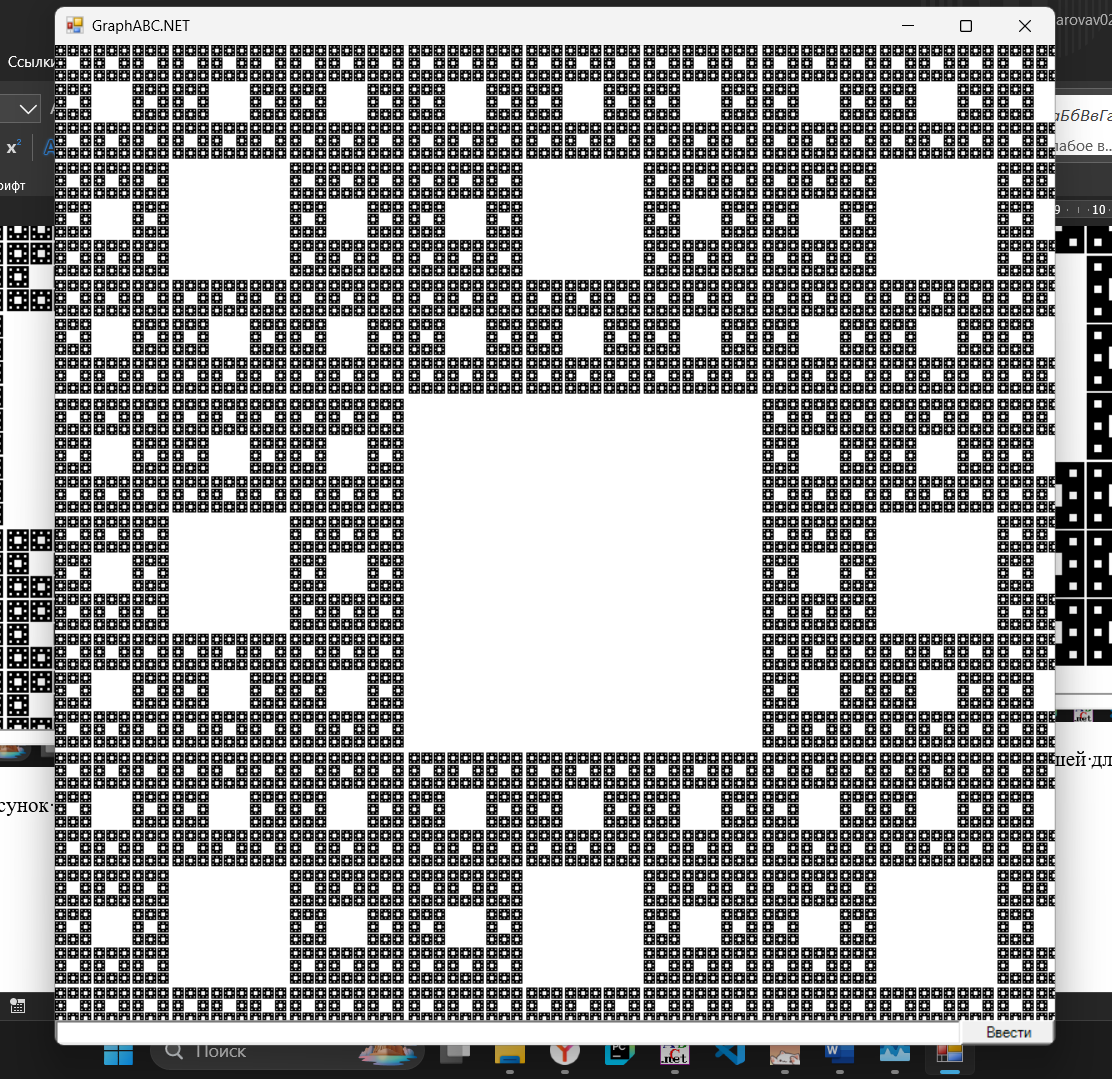


Рисунок 5 – ковер Серпинского с большей длиной и глубиной рекурсии

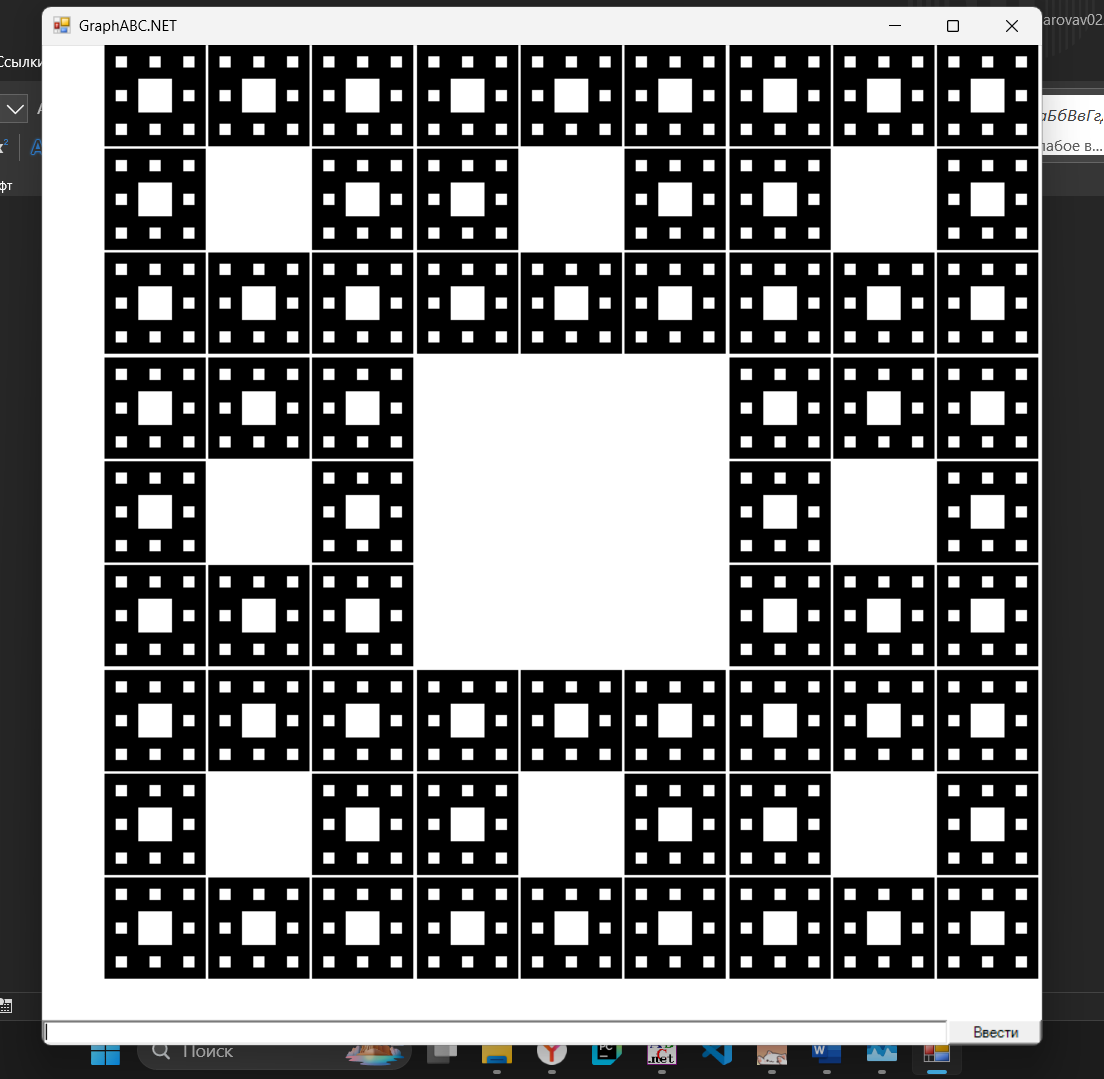


Рисунок 6 – сдвиг вправо

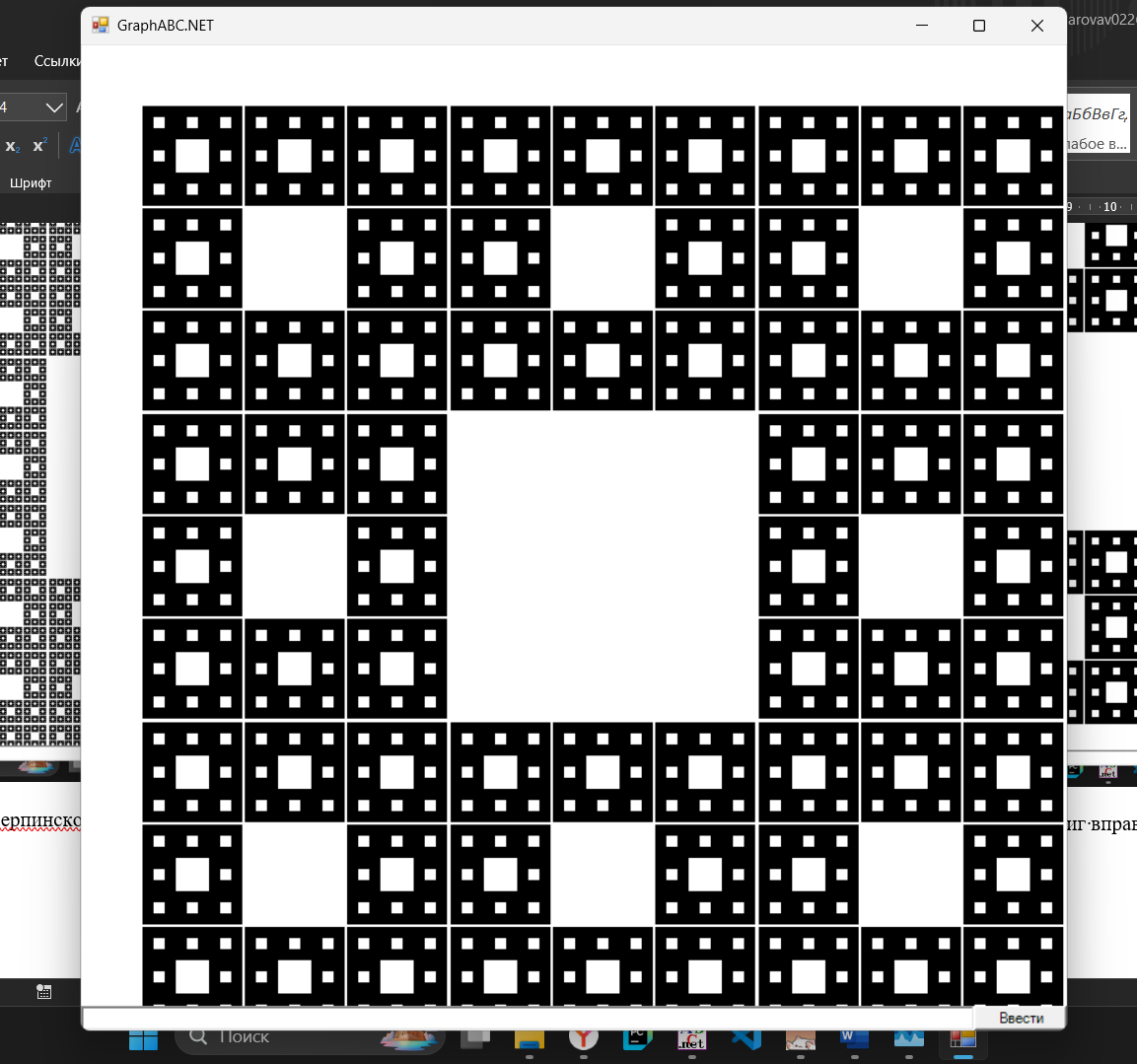


Рисунок 7 – сдвиг вниз

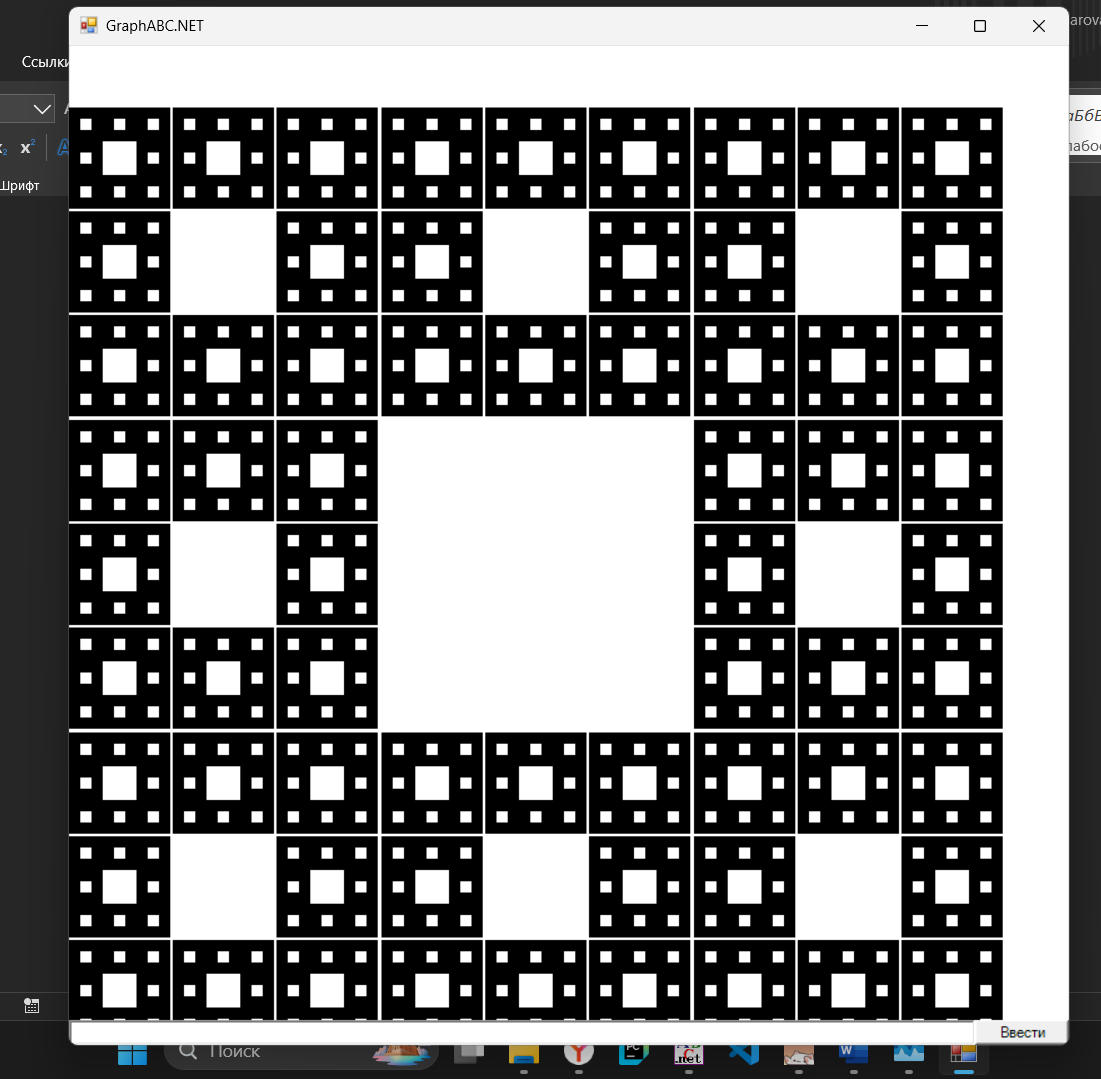


Рисунок 8 – сдвиг влево

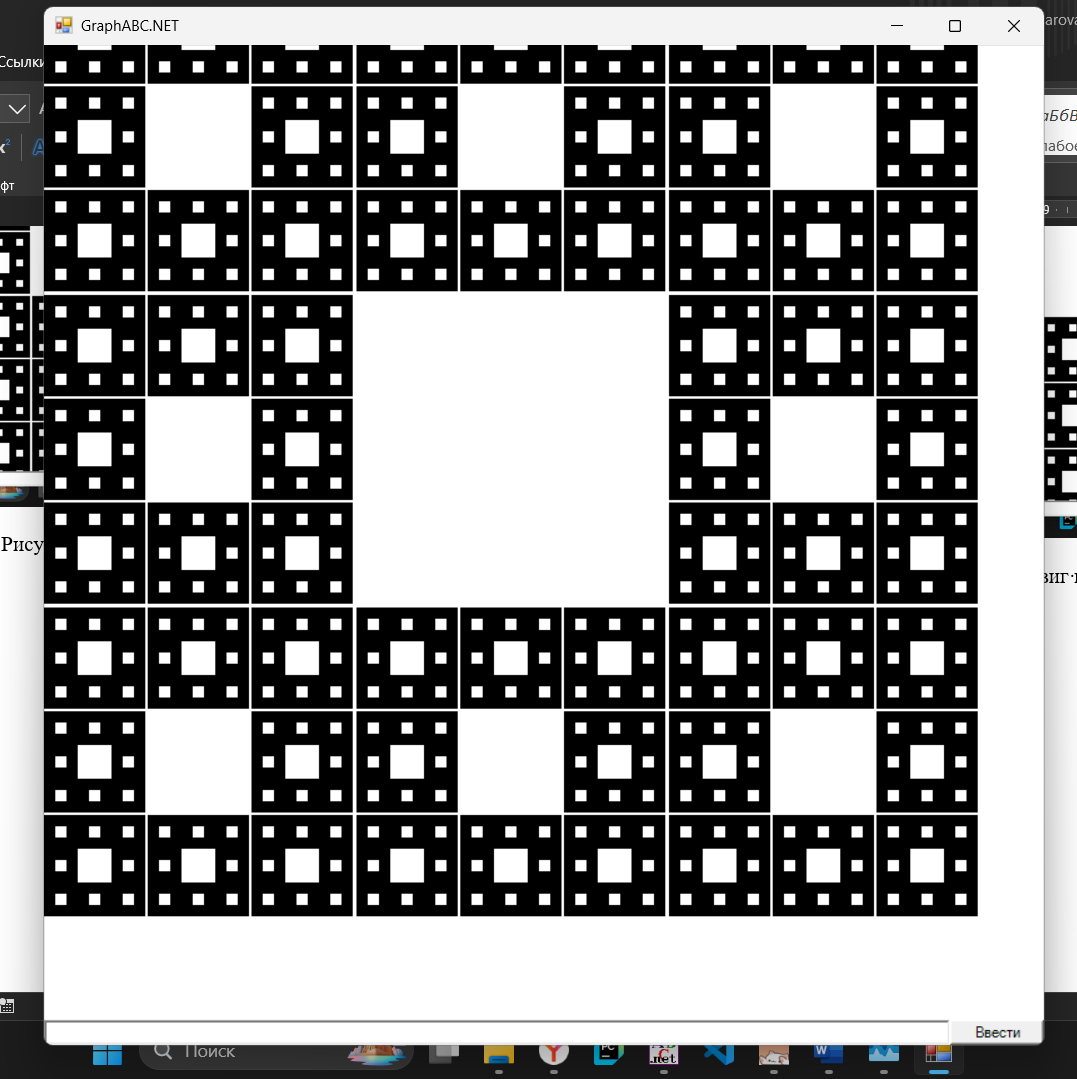


Рисунок 9 – сдвиг вверх

**7. ВЫВОД**

Фракталы представляют собой сложные структуры, которые обладают свойством самоподобия, то есть они выглядят одинаково на разных масштабах.

Ковер Серпинского — это двумерный фрактал, который создается с помощью рекурсивного деления квадрата на меньшие квадраты.

Исследование фракталов и их визуализация — это увлекательная область, которая сочетает в себе математику и искусство. Ковер Серпинского является ярким примером фрактала, который можно легко реализовать с помощью рекурсивных алгоритмов. Использование языка программирования Паскаль позволяет эффективно создавать графические представления таких сложных структур. Фракталы не только красивы визуально, но и имеют глубокие математические свойства и практическое применение в различных областях науки и техники.