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-206-52-00

Шерстобитов Михаил Ромнаович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

1. Цель работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма и ответы на вопросы
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод

1. Цель работы: получение навыков для визуализации алгоритмов с рекурсивным вычислениями, знакомство с фракталами

2.Формулировка заданий:

Вариант №13

Задание.

1. Написать программу для визуализации фрактала "ковер Серпинского".
2. Предусмотреть возможности масштабирования, изменения глубины прорисовки и перемещения полученной фигуры.
3. Построение множества КВДРАТОВ образующих фрактал, должно осуществляться в отдельном модуле.

3. Описание алгоритма и ответы на вопросы.

1. **Загрузка модулей**:

Программа использует модули GraphABC для работы с графикой.

1. **Инициализация переменных**:

1) depth — глубина рекурсии (начальное значение 5).

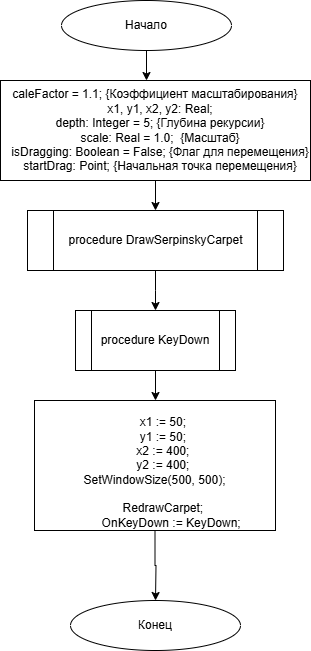
2) scale — масштаб (начальное значение 1.0).

1. **Создание окна**:

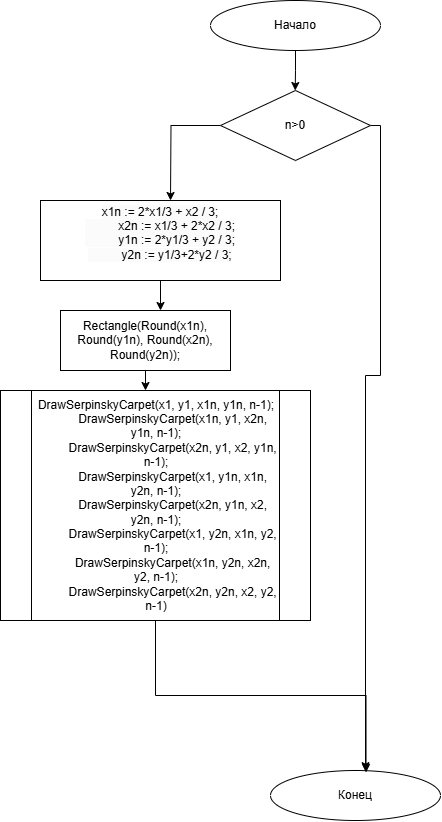
Устанавливается размер окна (500x500) и заголовок

4. Схема алгоритма

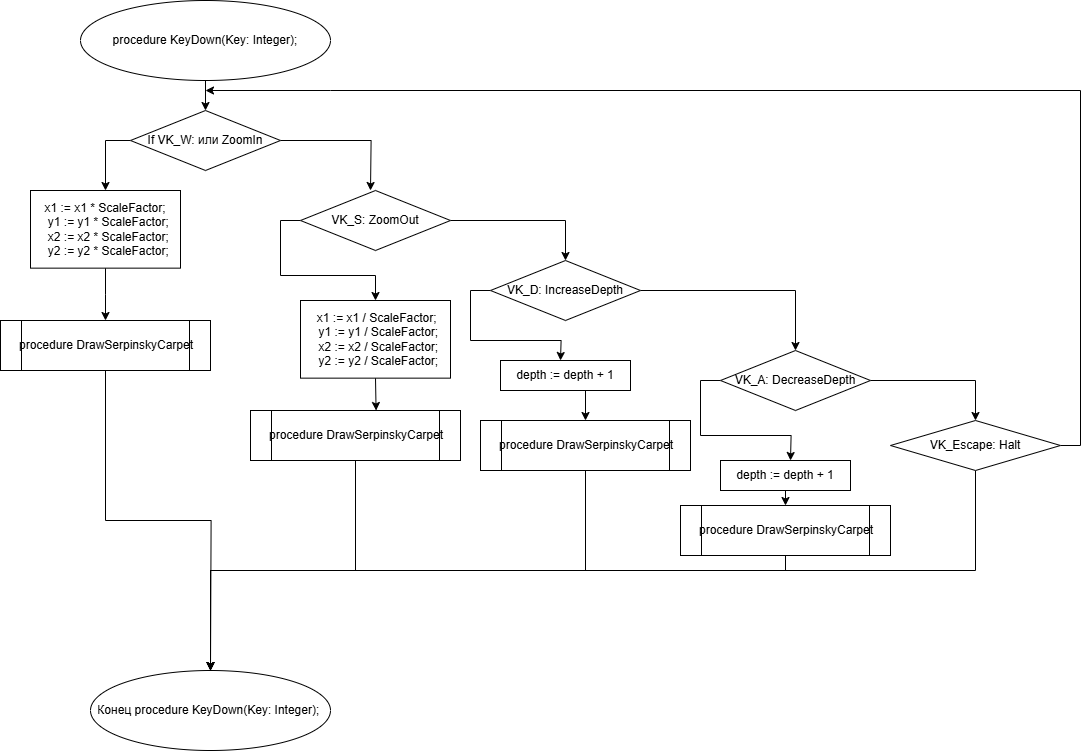
Работа Основной программы



Модуль ковра серпинского (DrawSerpinskyCarpet)



KeyDown и case выбора кнопок



1. Код программы

МОДУЛЬ

**procedure** DrawSerpinskyCarpet(x1, y1, x2, y2: Real; n: Integer);

**implementation**

**procedure** DrawSerpinskyCarpet(x1, y1, x2, y2: Real; n: Integer); **п**араметризация

**var**

x1n, y1n, x2n, y2n: Real;

**begin**

**if** n > 0 **then** база рекурсии

**begin**

x1n := 2\*x1/3 + x2 / 3;

x2n := x1/3 + 2\*x2 / 3;

y1n := 2\*y1/3 + y2 / 3;

y2n := y1/3+2\*y2 / 3;

Rectangle(Round(x1n), Round(y1n), Round(x2n), Round(y2n));

DrawSerpinskyCarpet(x1, y1, x1n, y1n, n-1);

DrawSerpinskyCarpet(x1n, y1, x2n, y1n, n-1);

DrawSerpinskyCarpet(x2n, y1, x2, y1n, n-1);

DrawSerpinskyCarpet(x1, y1n, x1n, y2n, n-1); декомпозиция

DrawSerpinskyCarpet(x2n, y1n, x2, y2n, n-1);

DrawSerpinskyCarpet(x1, y2n, x1n, y2, n-1);

DrawSerpinskyCarpet(x1n, y2n, x2n, y2, n-1);

DrawSerpinskyCarpet(x2n, y2n, x2, y2, n-1)

**end**

**end**;

**end**.

Вторая программа для логики генерации КОВРА СЕРПИНСОКГО

**uses** GraphABC, SerpinskyCarpet;

**Const**

ScaleFactor = 1.1; {Коэффициент масштабирования}

**Var**

x1, y1, x2, y2: Real;

depth: Integer = 5; {Глубина рекурсии}

scale: Real = 1.0; {Масштаб}

isDragging: Boolean = False; {Флаг для перемещения}

startDrag: Point; {Начальная точка перемещения}

{Процедура для перерисовки ковра Серпинского}

**procedure** RedrawCarpet;

**begin**

ClearWindow; {Очищаем окно}

DrawSerpinskyCarpet(x1, y1, x2, y2, depth); {Рисуем ковер}

Redraw; {Обновляем графическое окно}

**end**;

{Процедура для увеличения масштаба}

**procedure** ZoomIn;

**begin**

x1 := x1 \* ScaleFactor;

y1 := y1 \* ScaleFactor;

x2 := x2 \* ScaleFactor;

y2 := y2 \* ScaleFactor;

RedrawCarpet; {Перерисовываем ковер}

**end**;

{Процедура для уменьшения масштаба}

**procedure** ZoomOut;

**begin**

x1 := x1 / ScaleFactor;

y1 := y1 / ScaleFactor;

x2 := x2 / ScaleFactor;

y2 := y2 / ScaleFactor;

RedrawCarpet; {Перерисовываем ковер}

**end**;

{Процедура для увеличения глубины рекурсии}

**procedure** IncreaseDepth;

**begin**

depth := depth + 1; {Увеличиваем глубину}

RedrawCarpet; {Перерисовываем ковер}

**end**;

{Процедура для уменьшения глубины рекурсии}

**procedure** DecreaseDepth;

**begin**

**if** depth > 1 **then** {Глубина не может быть меньше 1}

depth := depth - 1; {Уменьшаем глубину}

RedrawCarpet; {Перерисовываем ковер}

**end**;

{Обработчик нажатия клавиш}

**procedure** KeyDown(Key: Integer);

**begin**

**case** Key **of**

VK\_W: ZoomIn; {Увеличиваем масштаб}

VK\_S: ZoomOut; {Уменьшаем масштаб}

VK\_D: IncreaseDepth; {Увеличиваем глубину рекурсии}

VK\_A: DecreaseDepth; {Уменьшаем глубину рекурсии}

VK\_Escape: Halt; {Выход по клавише Esc}

**end**;

**end**;

**Begin**

{Инициализация начальных координат}

x1 := 50;

y1 := 50;

x2 := 400;

y2 := 400;

{Инициализация графического режима}

SetWindowSize(500, 500);

LockDrawing; {Включаем режим отложенной отрисовки}

{Рисуем ковер Серпинского}

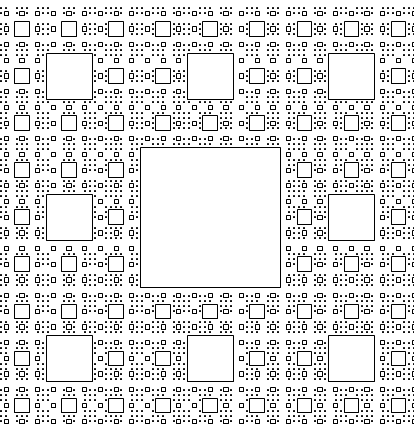
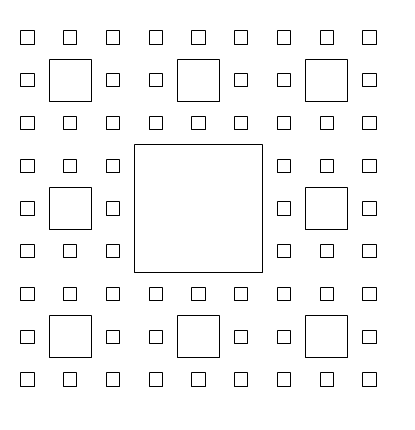
RedrawCarpet;

{Назначаем обработчик нажатия клавиш}

OnKeyDown := KeyDown;

**End**.

6. Результат выполнения программы

Вопросы:

1. Что такое фрактал -  [множество](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE), обладающее свойством [самоподобия](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%BC%D0%BE%D0%BF%D0%BE%D0%B4%D0%BE%D0%B1%D0%B8%D0%B5) (объект, в точности или приближённо совпадающий с частью себя самого, то есть целое имеет ту же форму, что и одна или более частей)(снежинка кроны деревьев).

2 Строится ковер Серпинского следующим образом. Вначале берётся квадрат со стороной равной единице, затем каждая сторона квадрата делится на три равные части, а весь квадрат, соответственно, на девять одинаковых квадратиков со стороной равной . Из полученной фигуры вырезается центральный квадрат.

7. Вывод.

Получил навыки для визуализации алгоритмов с рекурсивным вычислениями, знакомство с фракталами. При работе у меня возникло несколько проблем, при написании кода, а именно написании логики генерации КОВРА СЕРПИНСКОГО, размещение КВАДРАТОВ, образующих фрактал, отдельном модуле.