Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №1**

**«ИССЛЕДОВАНИЕ ФРАКТАЛОВ»**

**ПО ДИСЦИПЛИНЕ «РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ»**

Выполнил: студент учебной группы

ИСПк-207-52-00

Наумкин Глеб Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

Содержание

1. Цель лабораторной работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод
8. **Цель работы:** получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.
9. **Формулировка задания (Вариант:9)**

1. Написать программу для визуализации фрактала «Ковёр Серпинского».

2. Предусмотреть возможности масштабирования, изменения глубины прорисовки и перемещения полученной фигуры.

3. Построение множества ломанных, образующих фрактал должно осуществляться в отдельном модуле.

1. **Описание алгоритма**

***Задание № 1***

1. **Устанавливаем начальные значения переменных: DEPTH равно 5, FractalX равно 50, FractalY равно 50, FractalSize равно 500.**
2. **Устанавливаем заголовок окна на "Фрактал Ковер Серпинского" и размер окна 800х800.**
3. **Рисуем фрактал Ковер Серпинского с начальными параметрами FractalX, FractalY, FractalSize и DEPTH.**

***Задание № 2***

1. Устанавливаем константы: MOVE\_STEP равно 10, ZOOM\_STEP равно 20, START\_SIZE равно 500.
2. Создаем процедуру KeyDown(Key: Integer) для обработки событий клавиатуры:
   * Если нажата клавиша влево (VK\_LEFT), уменьшаем FractalX на MOVE\_STEP.
   * Если нажата клавиша вправо (VK\_RIGHT), увеличиваем FractalX на MOVE\_STEP.
   * Если нажата клавиша вверх (VK\_UP), уменьшаем FractalY на MOVE\_STEP.
   * Если нажата клавиша вниз (VK\_DOWN), увеличиваем FractalY на MOVE\_STEP.
   * Если нажата клавиша "минус" (VK\_SUBTRACT), уменьшаем FractalSize на ZOOM\_STEP.
   * Если нажата клавиша "плюс" (VK\_ADD), увеличиваем FractalSize на ZOOM\_STEP.
   * Очищаем окно.
   * Перерисовываем фрактал Ковер Серпинского с новыми параметрами.
3. Устанавливаем заголовок окна на "Фрактал Ковер Серпинского" и размер окна 800х800.
4. Устанавливаем обработчик событий клавиатуры на процедуру KeyDown.

***Задание № 3***

1. Определить константу depth равную 5 - глубину прорисовки.
2. Создать процедуру DrawCover с параметрами x, y, size, depth:
   * Если depth равен 0, то выйти из процедуры.
   * Иначе:
     + Установить цвет ручки как черный.
     + Нарисовать прямоугольник с координатами x, y и размером size.
     + Уменьшить size на треть.
     + Рекурсивно вызвать DrawCover со следующими параметрами:
       - (x, y, size, depth - 1)
       - (x + size, y, size, depth - 1)
       - (x + 2 \* size, y, size, depth - 1)
       - (x, y + size, size, depth - 1)
       - (x + 2 \* size, y + size, size, depth - 1)
       - (x, y + 2 \* size, size, depth - 1)
       - (x + size, y + 2 \* size, size, depth - 1)
       - (x + 2 \* size, y + 2 \* size, size, depth - 1)
3. Создать процедуру DrawCovrik с параметрами x, y, size, depth, которая вызывает DrawCover(x, y, size, depth).
4. Выполнить DrawCovrik с заданными параметрами для отображения фрактала Covrik.
5. **Схема алгоритма с комментариями (доделать схему где Exit стрелку в конец и переход схемы с помощью кружочка, также DrawCover в квадрате сделать с палочками как в начале.)**

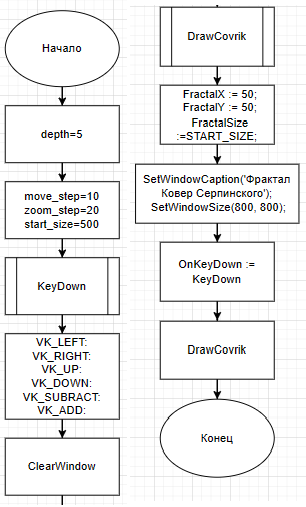


Рис 1. Схема алгоритма (задание 1,2) с комментариями.

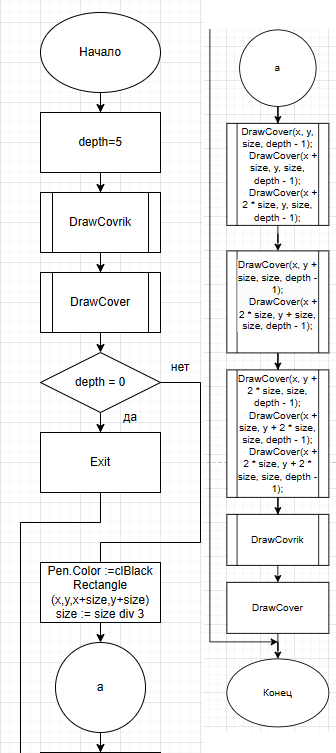


Рис 1. Схема алгоритма (задание 3) с комментариями.

1. **Код программы**
2. **unit** FractalModule;
3. **interface**
4. **uses** graphABC;
5. **const**
6. depth = 5; // глубина прорисовки
7. **procedure** DrawCovrik(x, y, size, depth: integer);
8. **implementation**
9. **procedure** DrawCover(x, y, size, depth: integer); // процедура для рекурсивного построения фрактала
10. **begin**
11. **if** depth = 0 **then**
12. **exit**
13. **else**
14. **begin**
15. Pen.Color := clBlack;
16. Rectangle(x, y, x + size, y + size);
17. size := size **div** 3;
18. DrawCover(x, y, size, depth - 1);
19. DrawCover(x + size, y, size, depth - 1);
20. DrawCover(x + 2 \* size, y, size, depth - 1);
22. DrawCover(x, y + size, size, depth - 1);
23. DrawCover(x + 2 \* size, y + size, size, depth - 1);
25. DrawCover(x, y + 2 \* size, size, depth - 1);
26. DrawCover(x + size, y + 2 \* size, size, depth - 1);
27. DrawCover(x + 2 \* size, y + 2 \* size, size, depth - 1);
28. **end**;
29. **end**;
30. **procedure** DrawCovrik(x, y, size, depth: integer);
31. **begin**
32. DrawCover(x, y, size, depth);
33. **end**;
34. **end**.

**uses** FractalModule, graphABC, Controls;

**var** DEPTH: Integer = 5; // Глубина фрактала

FractalX, FractalY, FractalSize: Integer;

**const** MOVE\_STEP = 10;

**const** ZOOM\_STEP = 20;

**const** START\_SIZE = 500;

// Обработчик событий клавиатуры

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

**begin**

**case** Key **of**

VK\_LEFT: FractalX := FractalX - MOVE\_STEP; // Продвигаемся влево

VK\_RIGHT: FractalX := FractalX + MOVE\_STEP; // Продвигаемся вправо

VK\_UP: FractalY := FractalY - MOVE\_STEP; // Продвигаемся вверх

VK\_DOWN: FractalY := FractalY + MOVE\_STEP; // Продвигаемся вниз

VK\_SUBTRACT: FractalSize := FractalSize - ZOOM\_STEP; // Уменьшаем размер

VK\_ADD: FractalSize := FractalSize + ZOOM\_STEP; // Увеличиваем размер

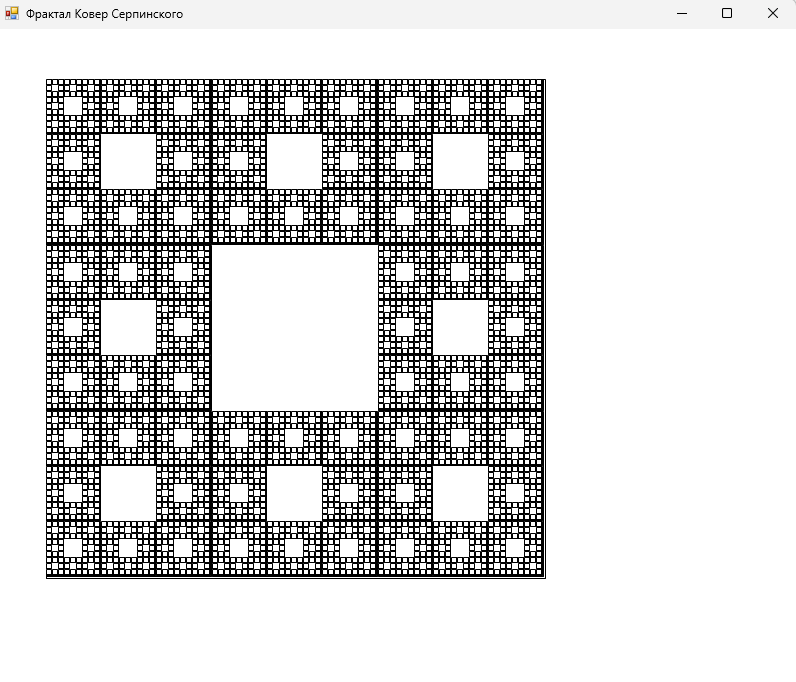
**end**;

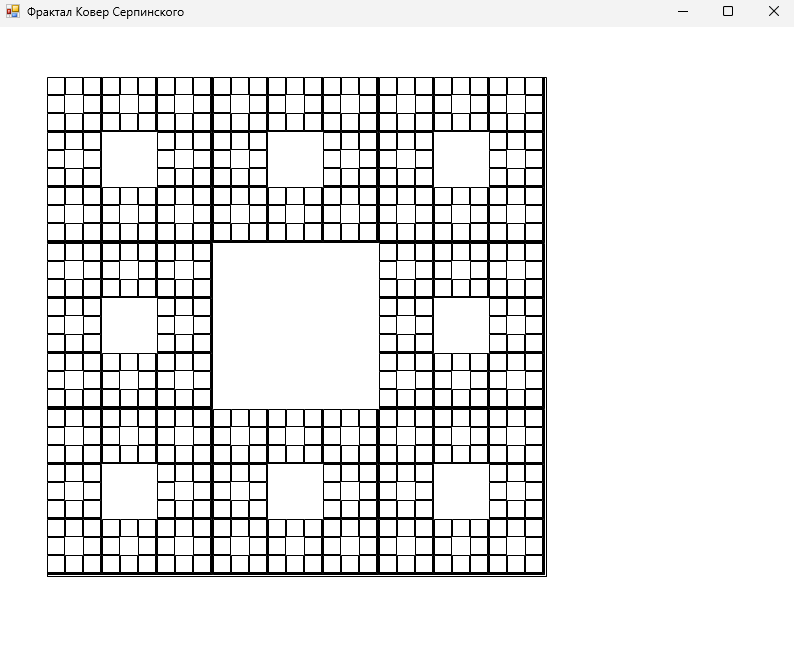
ClearWindow; // Очищаем окно перед перерисовкой

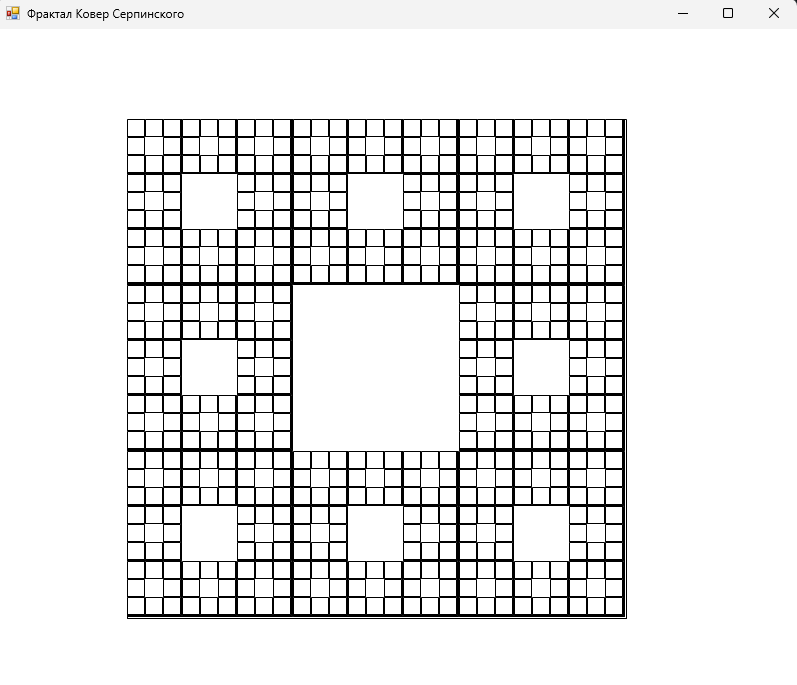
DrawCovrik(FractalX, FractalY, FractalSize, DEPTH); // Перерисовываем с новыми параметрами

**end**;

**5. Результат выполнения программы**

******





**Вывод**

Таким образом мы получили навыки реализации алгоритмов с рекурсивными вычислениями, и познакомились с фракталами.