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 ДИСЦИПЛИНЕ «**МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Обухов Илья Николаевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**Задание:**

Вариант 17

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

**Описание и схема алгоритма:**

Ковер Серпинского – берется квадрат, разбивается на девять равных квадратов, средний из которых выбрасывается, а с остальными повторяется та же операция до бесконечности.

Алгоритм:

1) Построить квадрат

2) Разделить квадрат на 9 равных частей путем добавления двух вертикальных и горизонтальных линий.

3) Удалить центральный квадрат.

4) Для каждого из оставшихся восьми квадратов повторить шаги 2 и 3 (рекурсивно) до тех пор, пока не достигнется желаемая глубина.

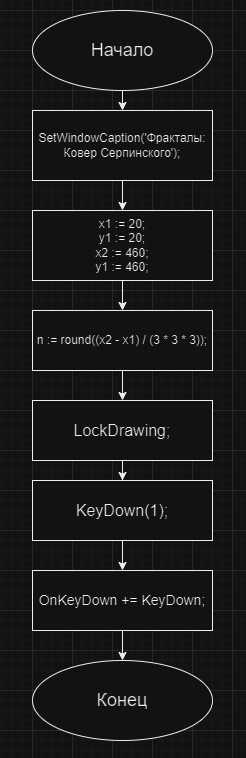


Рис.1 Процедура SetWindowCaption

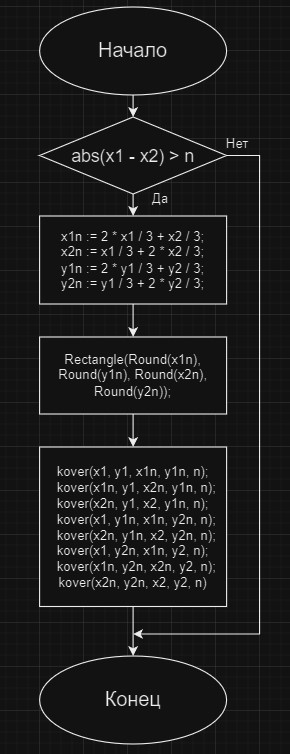


Рис.2 Процедура kover

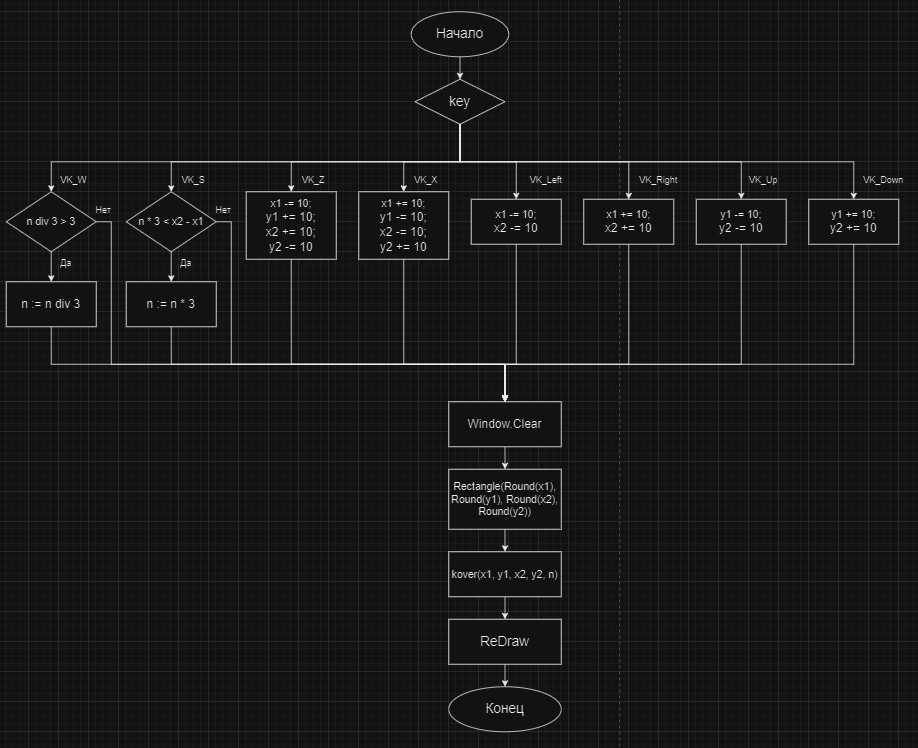


Рис.3 Процедура KeyDown

**Код программы:**

Главный модуль:

**uses** GraphABC, draw\_block;

**var**

n: Integer;

x1, y1, x2, y2: Real;

**procedure** KeyDown(key: integer);

**begin**

**case** key **of**

VK\_W: **if** n **div** 3 >3 **then** n := n **div** 3;

VK\_S: **if** n\*3 < x2-x1 **then** n := n\*3;

VK\_Z: **begin** x1-=10; y1+=10; x2+=10; y2-=10 **end**;

VK\_X: **begin** x1+=10; y1-=10; x2-=10; y2+=10 **end**;

VK\_Left: **begin** x1-=10; x2-=10 **end**;

VK\_Right: **begin** x1+=10; x2+=10 **end**;

VK\_Up: **begin** y1-=10; y2-=10 **end**;

VK\_Down: **begin** y1+=10; y2+=10 **end**;

**end**;

Window.Clear;

Rectangle(Round(x1), Round(y1), Round(x2), Round(y2));

kover(x1, y1, x2, y2, n);

ReDraw;

**end**;

**begin**

SetWindowCaption('Фракталы: Ковер Серпинского');

x1 := 20; y1 := 20; x2 := 460; y1 := 460; n := round((x2-x1) / (3\*3\*3));

LockDrawing;

KeyDown(1);

OnKeyDown += KeyDown;

**end**.

Модуль рисования:

**unit** draw\_block;

**uses** GraphABC;

**procedure** kover(x1, y1, x2, y2: Real; n: integer);

**var**

x1n, y1n, x2n, y2n: Real;

**begin**

**if** abs(x1-x2) > n **then**

**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));

kover(x1, y1, x1n, y1n, n);

kover(x1n, y1, x2n, y1n, n);

kover(x2n, y1, x2, y1n, n);

kover(x1, y1n, x1n, y2n, n);

kover(x2n, y1n, x2, y2n, n);

kover(x1, y2n, x1n, y2, n);

kover(x1n, y2n, x2n, y2, n);

kover(x2n, y2n, x2, y2, n)

**end**

**end**;

**begin**

**end**.

**Результат выполнения программы:**

**Изображение выглядит как прямоугольный, Прямоугольник, снимок экрана, диаграмма

Автоматически созданное описание**

Рис.4 – Ковер Серпинского.

Изображение выглядит как прямоугольный, Прямоугольник, диаграмма, снимок экрана

Автоматически созданное описание

Рис.5 – Перемещение Ковра

Изображение выглядит как диаграмма, Прямоугольник, прямоугольный, шаблон

Автоматически созданное описание

Рис.6 – Изменение масштабирования

Изображение выглядит как прямоугольный, Прямоугольник, диаграмма, шаблон

Автоматически созданное описание

Рис.7 – Изменение глубины прорисовки

**Вывод**

В ходе выполнения домашней контрольной работы №7 были получены базовые навыки реализации алгоритмов с рекурсивными вычислениями.

Работа была разделена на несколько этапов: цель, формулировка задания, написание кода программы, результат выполнения программы, вывод. Код программы был написан на языке программирования Pascal. Схема алгоритма была составлена на сайте www.drawio.com – это удобная платформа для создания подобных схем алгоритмов.

Подводя итог, можно сказать, что в данной работе была достигнута цель и были освоены новые способы выполнения заданий, а именно создание электронных схем алгоритмов. Работа проведена успешно, получен новый и интересный опыт.