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-203-52-00

Леушина Анна Станиславовна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**2. Формулировка задания (Вариант: 15)**

1. Написать программу для визуализации фрактала «Снежинка Коха».

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

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

**3. Схема алгоритма с комментариями**

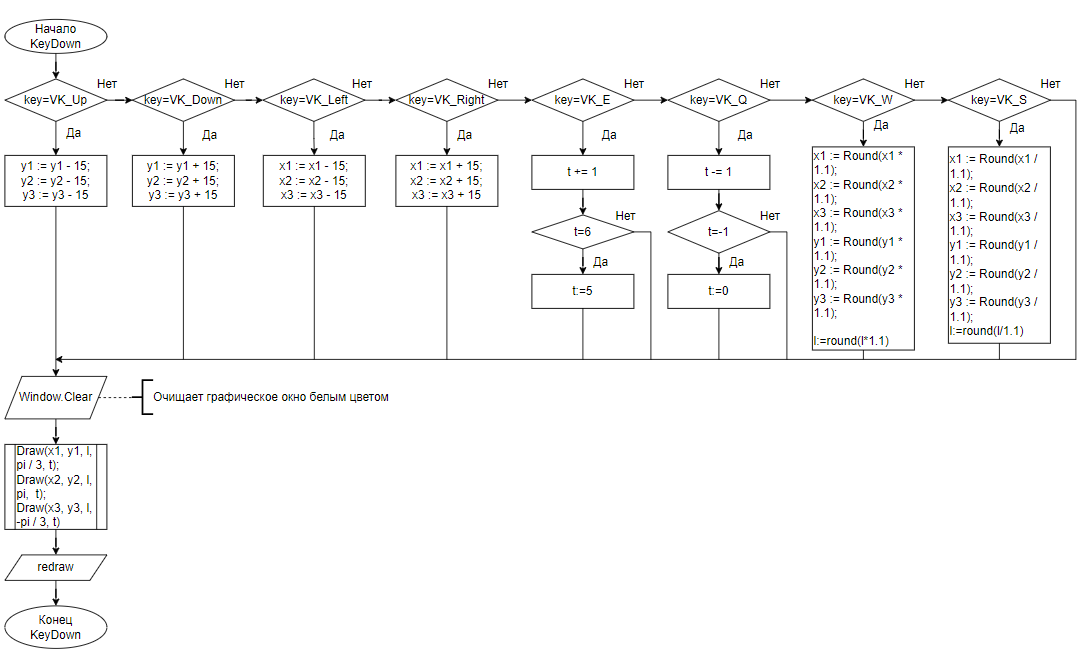


Рисунок 1 – Схема алгоритма procedure KeyDown

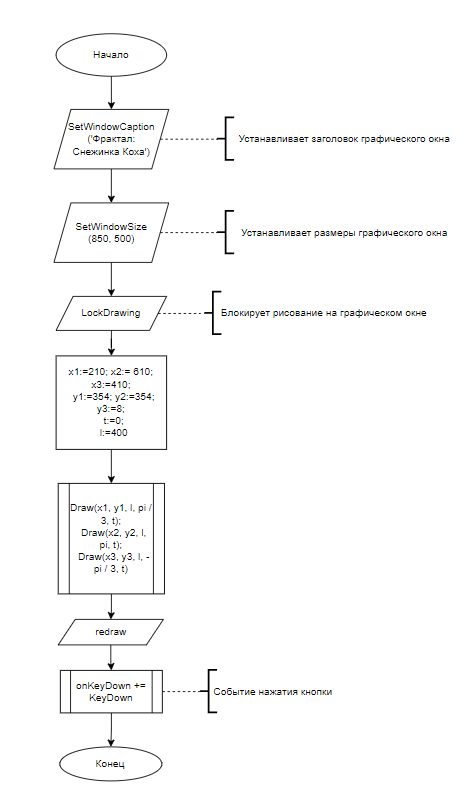


Рисунок 2 – Схема алгоритма

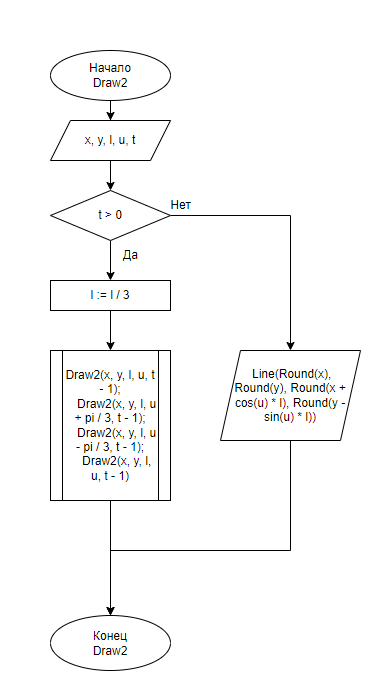


Рисунок 3 – Схема алгоритма procedure Draw2

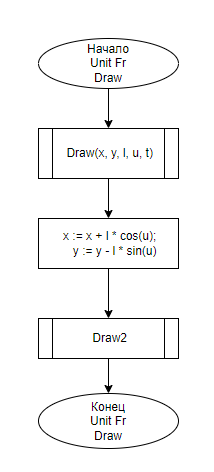


Рисунок 4 – Схема алгоритма Unit Fr procedure Draw

**4. Описание алгоритма**

**Фрактал снежинка Коха** – это геометрический фрактал, который строится путем замены каждого отрезка на определенную группу отрезков, таким образом, чтобы создавалась сложная структура, напоминающая снежинку. Фрактал Коха состоит из главного треугольника, каждая сторона которого заменяется на равносторонний треугольник.

**Алгоритм построения:**

* Задать начальную длину стороны треугольника (или отрезка) и начальную глубину рекурсии.
* Разделить сторону на три равные части.
* Построить четыре отрезка: два отрезка с длиной 1/3 и углом 60 градусов к исходной стороне, один отрезок с длиной 1/3 и углом -60 градусов, и один отрезок между концами этих трех отрезков.
* Уменьшить глубину рекурсии на 1 и повторить шаги 2-4 для каждого из трех новых отрезков.
* Продолжать рекурсивно уменьшать глубину и строить новые отрезки до тех пор, пока глубина не станет равной нулю.

**Ломанная генератор:**

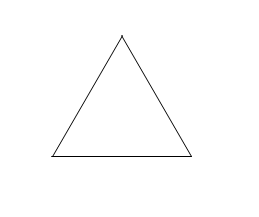


Рисунок 5 – Глубина 0

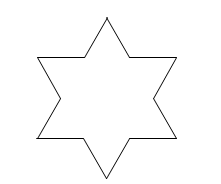


Рисунок 6 – Глубина 1

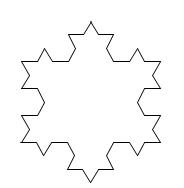


Рисунок 7 – Глубина 2

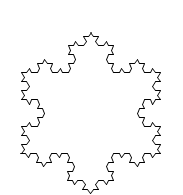


Рисунок 8 – Глубина 3

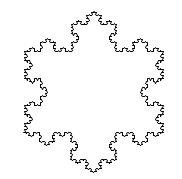


Рисунок 9 – Глубина 4

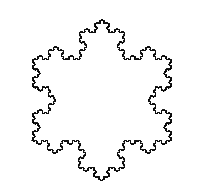


Рисунок 10 – Глубина 5

**5. Код программы:**

**Основная программа:**

**uses** GraphABC,Fr;

**var** x1, y1, x2, y2, x3, y3, l, t: integer;

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

**begin**

**case** key **of**

VK\_Up: **begin** y1 := y1 - 15; y2 := y2 - 15; y3 := y3 - 15 **end**;

VK\_Down: **begin** y1 := y1 + 15; y2 := y2 + 15; y3 := y3 + 15 **end**;

VK\_Left: **begin** x1 := x1 - 15; x2 := x2 - 15; x3 := x3 - 15 **end**;

VK\_Right: **begin** x1 := x1 + 15; x2 := x2 + 15; x3 := x3 + 15 **end**;

VK\_E: **begin** t += 1; **if** t=6 **then** t:=5; **end**;

VK\_Q: **begin** t -= 1; **if** t=-1 **then** t:=0; **end**;

VK\_W:

**begin**

x1 := Round(x1 \* 1.1);

x2 := Round(x2 \* 1.1);

x3 := Round(x3 \* 1.1);

y1 := Round(y1 \* 1.1);

y2 := Round(y2 \* 1.1);

y3 := Round(y3 \* 1.1);

l:=round(l\*1.1)

**end**;

VK\_S:

**begin**

x1 := Round(x1 / 1.1);

x2 := Round(x2 / 1.1);

x3 := Round(x3 / 1.1);

y1 := Round(y1 / 1.1);

y2 := Round(y2 / 1.1);

y3 := Round(y3 / 1.1);

l:=round(l/1.1)

**end**;

**end**;

Window.Clear;

Draw(x1, y1, l, pi / 3, t);

Draw(x2, y2, l, pi, t);

Draw(x3, y3, l, -pi / 3, t);

redraw;

**end**;

**begin**

SetWindowCaption('Фрактал: Снежинка Коха');

SetWindowSize(850, 500);

LockDrawing;

x1:=210; x2:= 610; x3:=410;

y1:=354; y2:=354; y3:=8;

t:=0;

l:=400;

Draw(x1, y1, l, pi / 3, t);

Draw(x2, y2, l, pi, t);

Draw(x3, y3, l, -pi / 3, t);

redraw;

onKeyDown += KeyDown;

**end**.

**Модуль:**

**Unit** Fr;

**Uses** GraphABC;

**procedure** Draw(x, y, l, u: Real; t: Integer);

**procedure** Draw2(**var** x, y: Real; l, u: Real; t: Integer);

**begin**

Draw(x, y, l, u, t);

x := x + l \* cos(u);

y := y - l \* sin(u);

**end**;

**begin**

**if** t > 0 **then**

**begin**

l := l / 3;

Draw2(x, y, l, u, t - 1);

Draw2(x, y, l, u + pi / 3, t - 1);

Draw2(x, y, l, u - pi / 3, t - 1);

Draw2(x, y, l, u, t - 1);

**end**

**else**

Line(Round(x), Round(y), Round(x + cos(u) \* l), Round(y - sin(u) \* l))

**end**;

**end**.

**6. Результат выполнения программы**

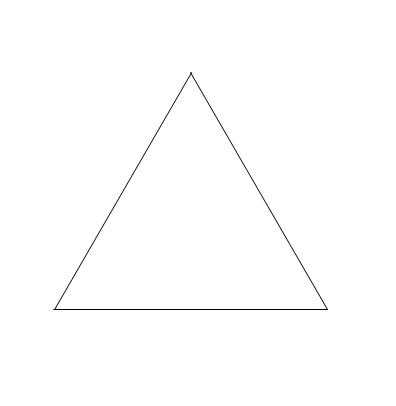
****

Рисунок 11 – Глубина 0

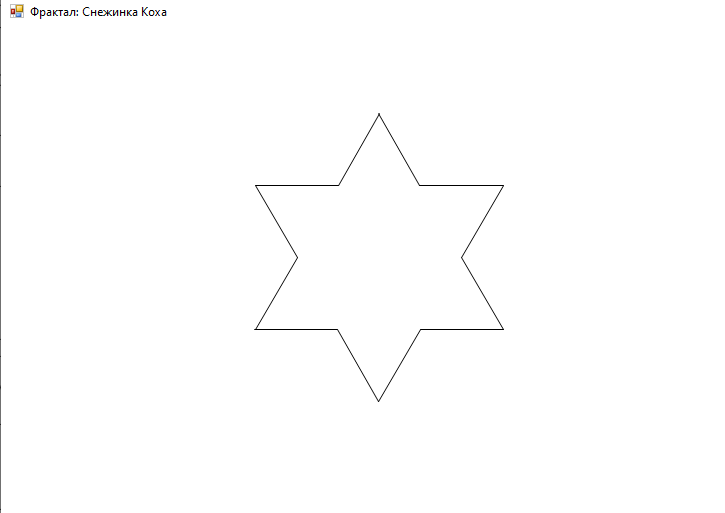
****

Рисунок 12 – Глубина 1

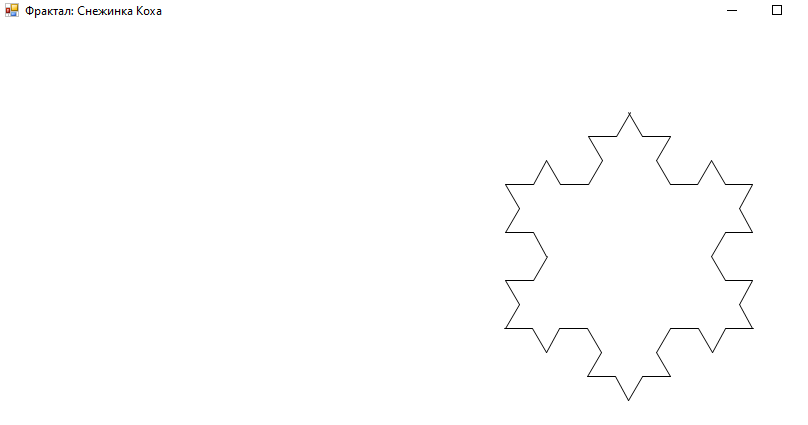


Рисунок 13 – Глубина 2, перемещение вправо

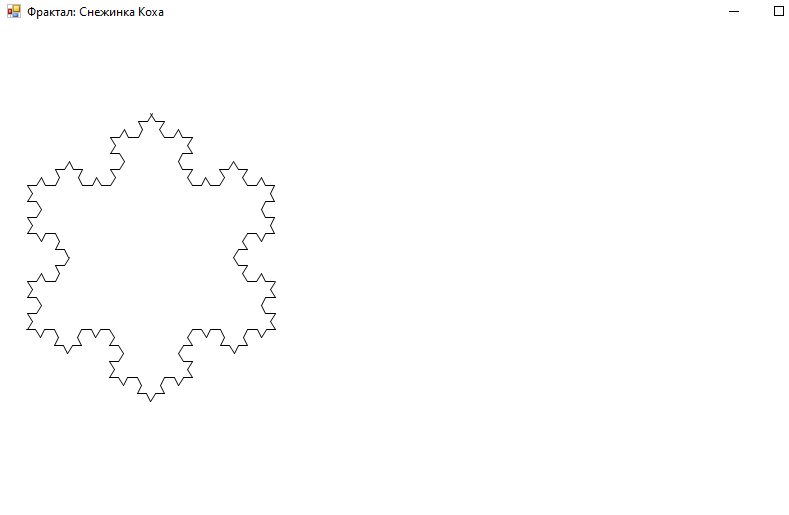


Рисунок 14 – Глубина 3, перемещение налево

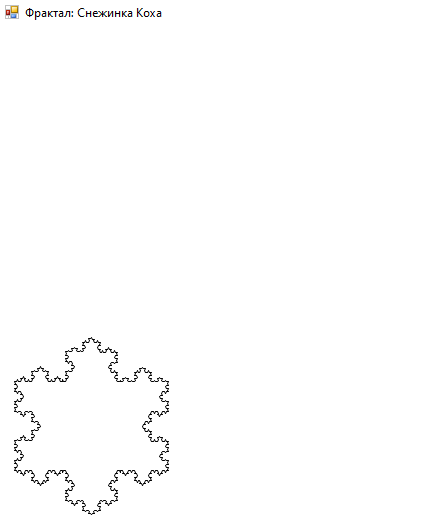


Рисунок 15– Глубина 4, перемещение вниз, уменьшение размера фрактала

**7. Вывод**

В ходе выполнения данной домашней контрольной работы были получены знания о визуализации фрактала, масштабировании, изменении глубины прорисовки и перемещения полученной фигуры. Так же были получены знания в создании модулей в языке программирования Pascal.

В работе была использована такая программа, как Draw.io. Draw.io - это онлайн-редактор диаграмм и схем, который позволяет создавать различные виды диаграмм, схем, планов и других графических объектов. Он широко используется в различных отраслях, таких как бизнес, образование, наука и т.д., для создания визуализаций, презентаций и документации. С помощью Draw.io можно легко и быстро создавать профессиональные диаграммы, схемы и другие графические объекты, а также экспортировать их в различные форматы, такие как PNG, JPEG, SVG и другие.