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

Выполнил: студент учебной группы

ИСПк-203-52-00 Казанцев Андрей Игоревич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**Задание:**

Вариант 10

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.

**Описание и схема алгоритма:**

Алгоритм построения снежинки Коха в этом коде работает следующим образом:

**Процедура Draw**: Эта процедура рисует линию на экране. Если глубина больше 0, то эта процедура вызывает себя четыре раза, каждый раз изменяя угол и длину линии. Если глубина равна 0, то рисуется прямая линия.

**Процедура Redraw:** Эта процедура очищает окно и вызывает процедуру Draw три раза, чтобы нарисовать три стороны снежинки Коха.

**Обработка нажатий клавиш:** В зависимости от нажатой клавиши изменяются параметры масштаба, глубины и смещения, а затем вызывается процедура Redraw для перерисовки снежинки Коха.

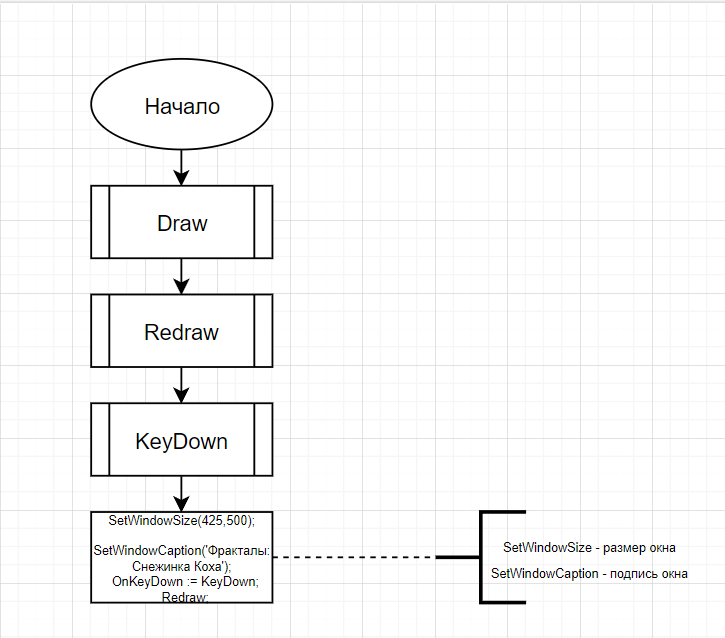
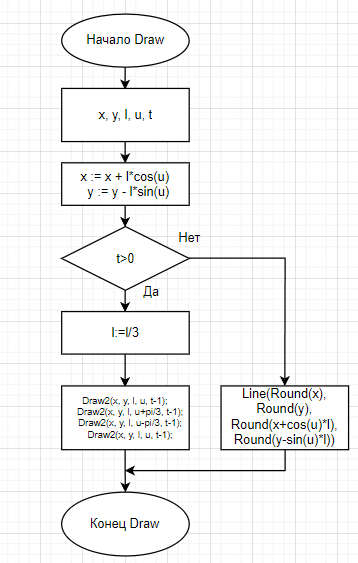


Рис.1 Общая схема.



Line – прорисовка линий, Round – округление значения

Рис.2 Процедура Draw

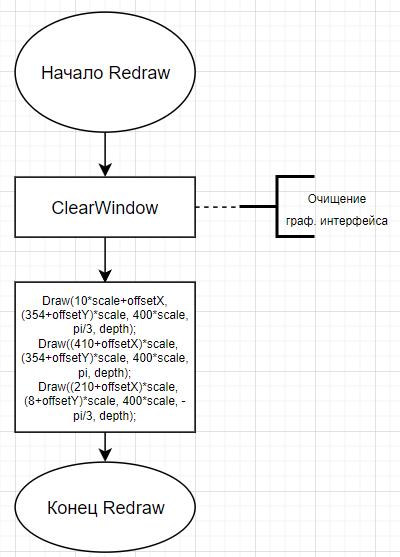


Рис.3 Процедура Redraw

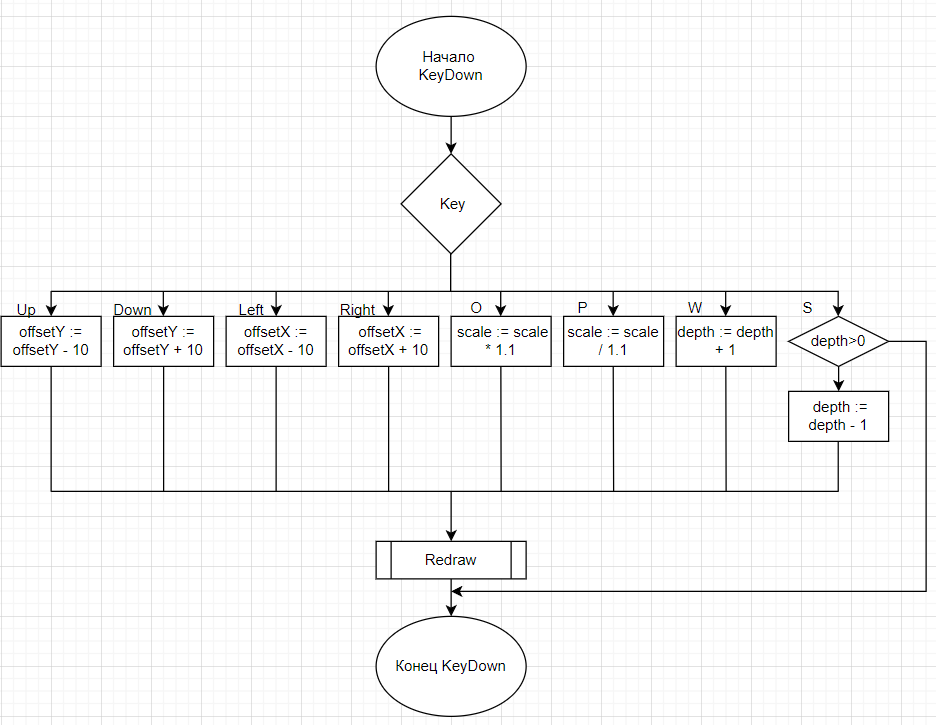
****

Рис.4 Процедура KeyDown

**Код программы:**

**main**

**uses** GraphABC, Fractal;

**var**

scale: Real = 1.0;

depth: Integer = 4;

offsetX: Real = 0.0;

offsetY: Real = 0.0;

**procedure** Redraw;

**begin**

ClearWindow;

Draw(10\*scale+offsetX, (354+offsetY)\*scale, 400\*scale, pi/3, depth);

Draw((410+offsetX)\*scale, (354+offsetY)\*scale, 400\*scale, pi, depth);

Draw((210+offsetX)\*scale, (8+offsetY)\*scale, 400\*scale, -pi/3, depth);

**end**;

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

**begin**

**case** Key **of**

VK\_Up: **begin**

offsetY := offsetY - 10;

Redraw;

**end**;

VK\_Down: **begin**

offsetY := offsetY + 10;

Redraw;

**end**;

VK\_Left: **begin**

offsetX := offsetX - 10;

Redraw;

**end**;

VK\_Right: **begin**

offsetX := offsetX + 10;

Redraw;

**end**;

VK\_O: **begin**

scale := scale \* 1.1;

Redraw;

**end**;

VK\_P: **begin**

scale := scale / 1.1;

Redraw;

**end**;

VK\_W: **begin**

depth := depth + 1;

Redraw;

**end**;

VK\_S: **if** depth > 0 **then begin**

depth := depth - 1;

Redraw;

**end**;

**end**;

**end**;

**begin**

SetWindowSize(425,500);

SetWindowCaption('Фракталы: Снежинка Коха');

OnKeyDown := KeyDown;

Redraw;

**end**.

**Fractal**

**unit** Fractal;

**interface**

**uses** GraphABC;

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

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

**implementation**

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

**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**;

**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**;

**end**.

**Результат выполнения программы:**

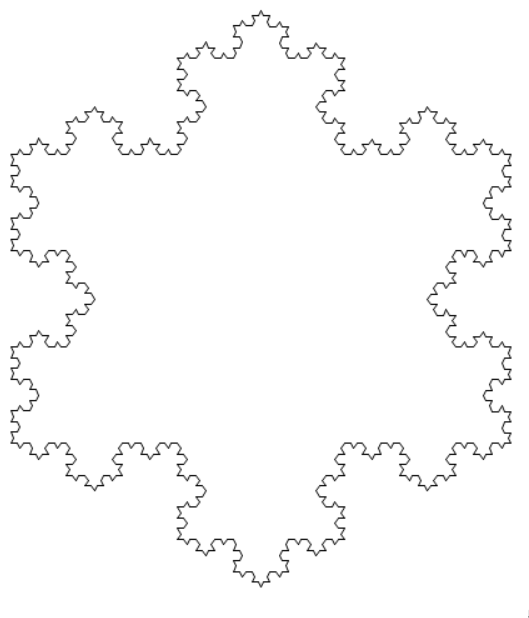
****

Рис.5 – Снежинка Коха.

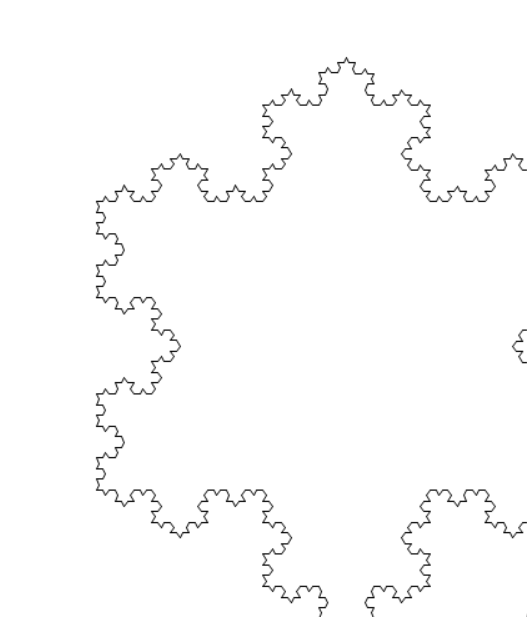


Рис.6 – Перемещение снежинки

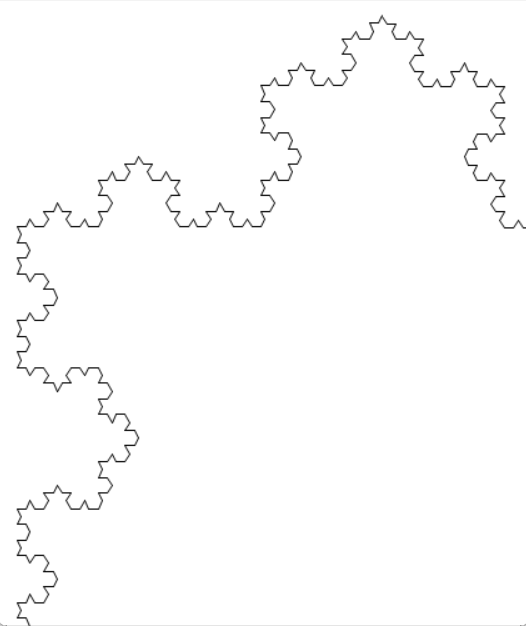


Рис.7 – Изменение масштабирования

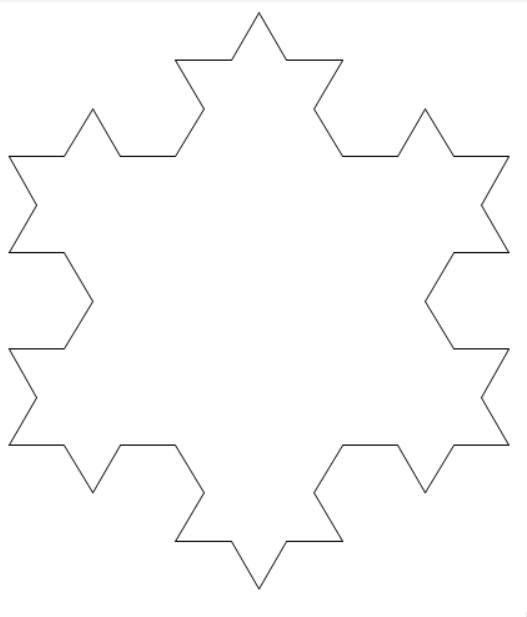


Рис.8 – Изменение глубины прорисовки

**Вывод**

Цель данной работы заключалась в том, чтобы получить навыки реализации алгоритмов с рекурсивными вычислениями, ознакомиться с фракталами. Эта цель была осуществима с помощью выполнения задания на создание фрактала и добавления возможности масштабирования, изменения глубины прорисовки и перемещения полученной фигуры. Данное задание отличалось количеством используемых процедур и функций. Тем самым оно вызывало больший интерес, т.к., выполняя, эти задания можно было столкнуться с неизвестными ранее функциями и процедурами в языке программирования Паскаль, такими как: Draw, Redraw, VK\_... и др. Данные функции были изучены и применены в разработке программы для нахождения значения функции.

Не менее интересными были результаты выполнения программы. При запуске кода, программа рисует фрактал Снежинку Коха, а при нажатии на определённые клавиши можно менять масштаб, глубину прорисовки снежинки и двигать фрактал.

Говоря, о написании программы и анализа результата её выполнения, нельзя не упомянуть о составлении схемы алгоритма, благодаря которой в последствие была написана программа. Схема алгоритма была составлена на сайте www.drawio.com – это удобная платформа для создания подобных схем алгоритмов. Составить здесь схему алгоритмов сможет любой человек, который когда-либо составлял их на бумаге. Работы получаются очень аккуратными и красивыми.

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 интересный и приятный опыт.