Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ЛАБОРАТОРНАЯ РАБОТА №7**

**«ИССЛЕДОВАНИЕ ФРАКТАЛОВ»**

**ПО ДИСЦИПЛИНЕ «МДК 05.02 Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-207-52-00

Перевозчиков Максим Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

Содержание

1. Цель работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы

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

**2.** **Формулировка задания (Вариант:4)**

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. Описание алгоритма**

Данный алгоритм позволяет интерактивно рисовать фрактал "Снежинка Коха" и управлять его параметрами с помощью клавиш на клавиатуре.

1. Процедура Draw рекурсивно рисует участок фрактала. Она принимает координаты начальной точки x, y, длину отрезка l, угол поворота u и глубину прорисовки t. Если глубина прорисовки больше нуля, то вызывается процедура Draw2, которая рисует четыре участка фрактала, повернутые на определенный угол.

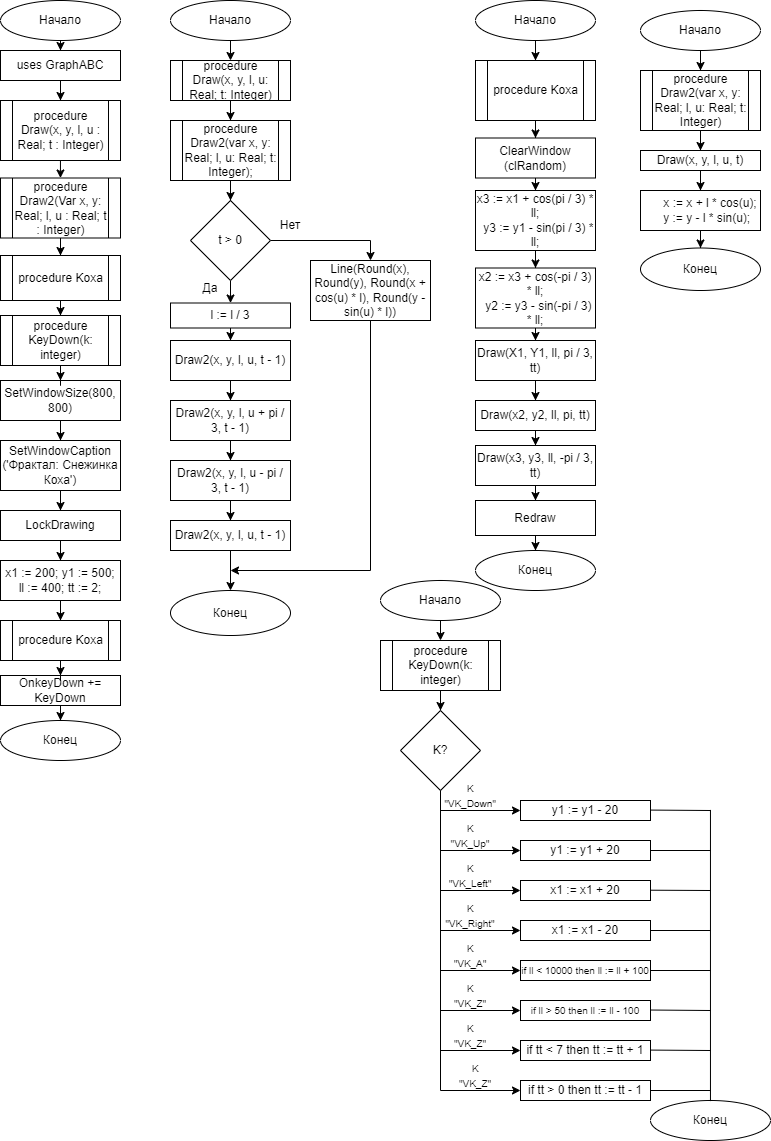
2. Процедура Koxa инициализирует начальные параметры фрактала (координаты начальной точки, длину отрезка и глубину прорисовки) и вызывает процедуру Draw для построения фрактала.

3. В процедуре KeyDown обрабатываются нажатия клавиш: стрелки вверх/вниз/влево/вправо перемещают фрактал по экрану, клавиши A и Z увеличивают/уменьшают масштаб фрактала, а клавиши S и X увеличивают/уменьшают глубину прорисовки.

4. В основном блоке программы устанавливаются параметры окна, вызывается процедура Koxa для отрисовки фрактала и устанавливается обработчик события нажатия клавиш.

1. **Схема алгоритма с описанием**

Блок схема к задаче



**5. Код программы**

**program** dkrfraktal;

**uses** GraphABC;

**var**

x1, y1, ll, tt: integer;

**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**;

**procedure** Koxa;

**var**

x2, y2, x3, y3: real;

**begin**

ClearWindow();

x3 := x1 + cos(pi / 3) \* ll;

y3 := y1 - sin(pi / 3) \* ll;

x2 := x3 + cos(-pi / 3) \* ll;

y2 := y3 - sin(-pi / 3) \* ll;

Draw(X1, Y1, ll, pi / 3, tt);

Draw(x2, y2, ll, pi, tt);

Draw(x3, y3, ll, -pi / 3, tt);

Redraw;

**end**;

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

**begin**

**case** K **of**

VK\_Down: y1 := y1 - 20; // передвижение вниз

VK\_Up: y1 := y1 + 20; // передвижение вверх

VK\_Left: x1 := x1 + 20; // передвижение влево

VK\_Right: x1 := x1 - 20; // передвижение вправо

VK\_A: **if** ll < 10000 **then** ll := ll + 100; //клавиша A = масштаб +

VK\_Z: **if** ll > 50 **then** ll := ll - 100; //клавиша Z = масштаб -

VK\_S: **if** tt < 7 **then** tt := tt + 1; //клавиша S = глубина прорисовки +

VK\_X: **if** tt > 0 **then** tt := tt - 1; //клавиша X = глубина прорисовки -

**end**;

Koxa;

**end**;

**begin**

SetWindowSize(800, 800);

SetWindowCaption('Фрактал: Снежинка Коха');

LockDrawing;

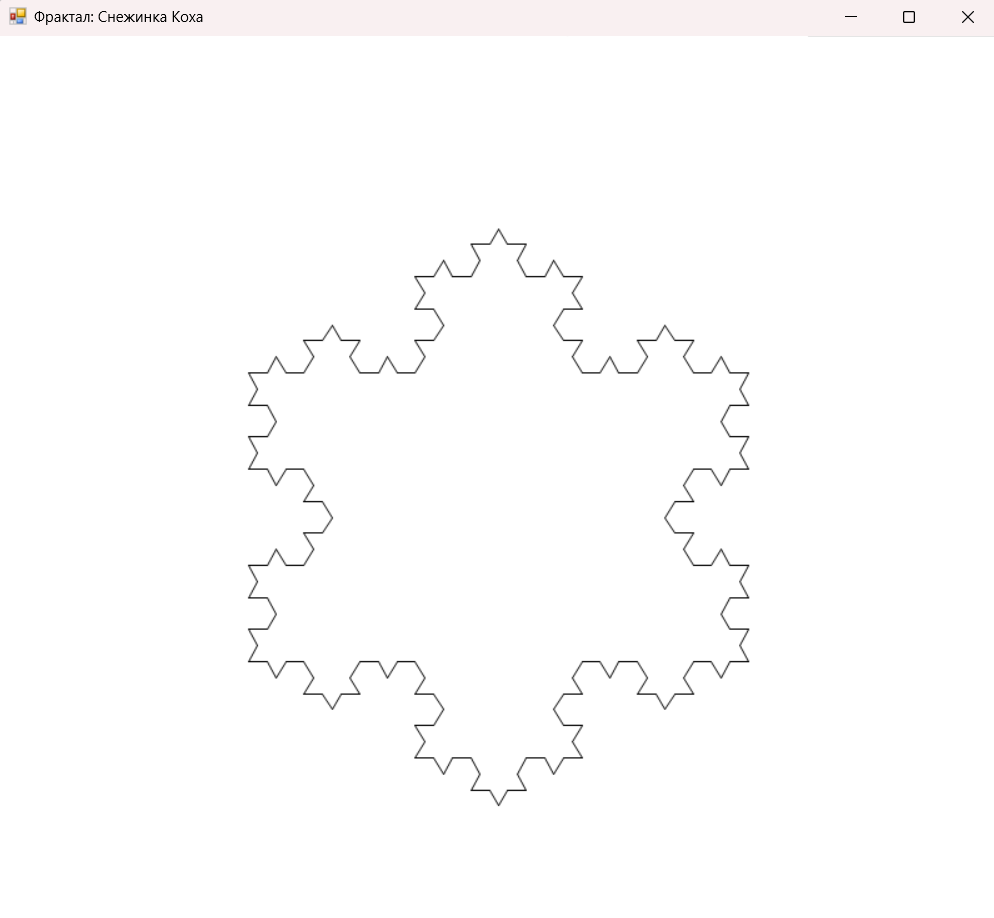
x1 := 200; y1 := 500; ll := 400; tt := 2;

Koxa;

OnkeyDown += KeyDown;

**end**.

**6.Результат выполнения программы**

****

**7.Вывод**

Данная домашняя работа была написана на языке программирования Pascal. В работе требовалось написать программу, которая строила фрактал “Снежинка Коха” с возможность глубины прорисовки и масштабирования.

Одной из трудностей данного кода может быть понимание рекурсивной логики построения фрактала "Снежинка Коха". Рекурсия используется для разбиения фигуры на более мелкие части и последующего их отрисовки. Понимание того, как каждый следующий уровень рекурсии формирует более сложную структуру фрактала, может быть непростой задачей.

Также, в данном коде используется обработка событий клавиатуры для управления параметрами фрактала. Это может быть непривычным для тех, кто не имеет опыта работы с обработкой событий в графических приложениях.

В процессе выполнения работы, для создания схем алгоритма для задач мы ознакомилась с программой draw.io. Draw.io – это инструмент для создания диаграмм, блок-схем, интеллект-карт и другого. Сервис распространяется на бесплатной основе с открытым исходным кодом. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя. Сервис позволяет отслеживать и восстанавливать изменения готовых продуктов, импортировать и экспортировать в PDF, PNG, XML, VSDX, HTML, а также автоматически публиковать и делиться работами. Инструмент работает с Google Диск, Google Workspace и Dropbox.