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»**

**«МДК 05. 02 Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-206-52-00

Никулин Е.В.

Преподаватель:

Сергеева Е.Г.

Киров

2024

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

**Задание**:

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 алгоритма**

**Инициализация:**

* Установка размера окна и заголовка.
* Блокировка отрисовки для предотвращения мерцания.
* Инициализация переменных x1, y1, ll и tt, которые определяют начальную позицию и параметры фрактала.
* Вызов процедуры snezhinka для отрисовки первоначального фрактала.
* Назначение обработчика событий OnKeyDown на функцию KeyDown.

**Процедура Draw:**

* Если t > 0, то рекурсивно вызывается Draw для отрисовки четырех фрагментов линий, которые образуют фрактал "Снежинка Коха".
* Каждый фрагмент отрисовывается на основе предыдущего фрагмента, уменьшая длину l в три раза и изменяя угол u на pi/3.
* Если t <= 0, то отрисовывается линия от точки (x, y) до точки, расположенной на расстоянии l от (x, y) в направлении u.

**Процедура Draw2:**

* Вызывает Draw для отрисовки фрагмента линии.
* Обновляет координаты x и y для следующего фрагмента, используя косинус и синус угла u и длины l.

**Процедура snezhinka:**

* Очищает окно.
* Вычисляет координаты вершин треугольника, образующего основание фрактала "Снежинки Коха".
* Вызывает Draw для отрисовки фрагментов фрактала на каждой из вершин.
* Перерисовывает окно.

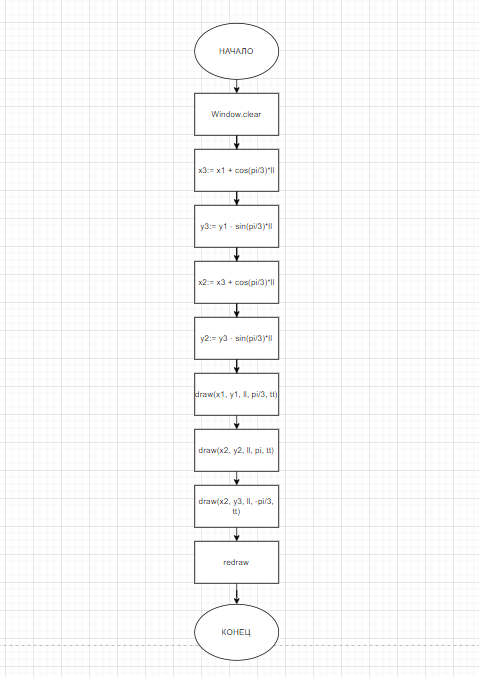
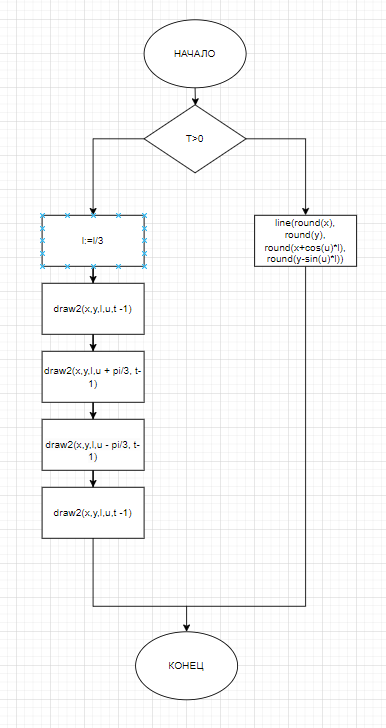
**Процедура KeyDown:**

* Обрабатывает нажатия клавиш:
* Стрелки вверх, вниз, влево, вправо: изменяют положение фрактала.
* Клавиша Escape: завершает работу программы.
* Клавиша A: увеличивает масштаб фрактала.
* Клавиша Z: уменьшает масштаб фрактала.
* Клавиша S: увеличивает количество итераций фрактала.
* Клавиша X: уменьшает количество итераций фрактала.

**Основная программа:**

* Запускается цикл обработки событий, ожидая нажатия клавиш.
* При нажатии клавиш вызывается процедура KeyDown, которая обновляет параметры фрактала и перерисовывает его.

**Схема алгоритма**

****

**Код программы**

**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** snezhinka;

**var** x2,y2,x3,y3: real;

**begin**

Window.Clear;

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+10; //курсоры в четыре стороны

VK\_Up: y1 := y1-10;

VK\_Left: x1 := x1-10;

VK\_Right: x1 := x1+10;

VK\_Escape: halt(1); //выход

VK\_A: **if** ll < 1000 **then** ll := ll+10; //клавиша A = масштаб +

VK\_Z: **if** ll > 50 **then** ll := ll-10; //клавиша Z = масштаб -

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

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

**end**;

snezhinka;

**end**;

**begin**

SetWindowSize(800,800);

SetWindowCaption('Фрактал: Снежинка Коха');

LockDrawing;

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

snezhinka;

OnkeyDown += KeyDown;

**end**.

**Результаты выполнения программы**

