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 ФРАКТАЛА»**

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

Выполнил: студент учебной группы

ИСПк-101-51-00

Бекмансуров Рустам Мхадтисович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**Формулировка задания:**

**Вариант 22**

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

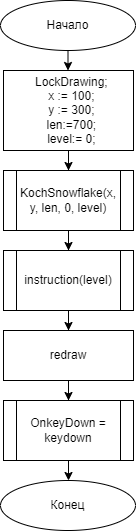


Рисунок 1 – алгоритм программы

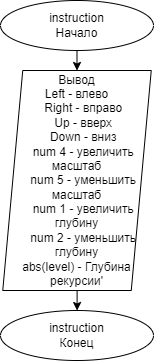


Рисунок 2 – алгоритм процедуры instruction

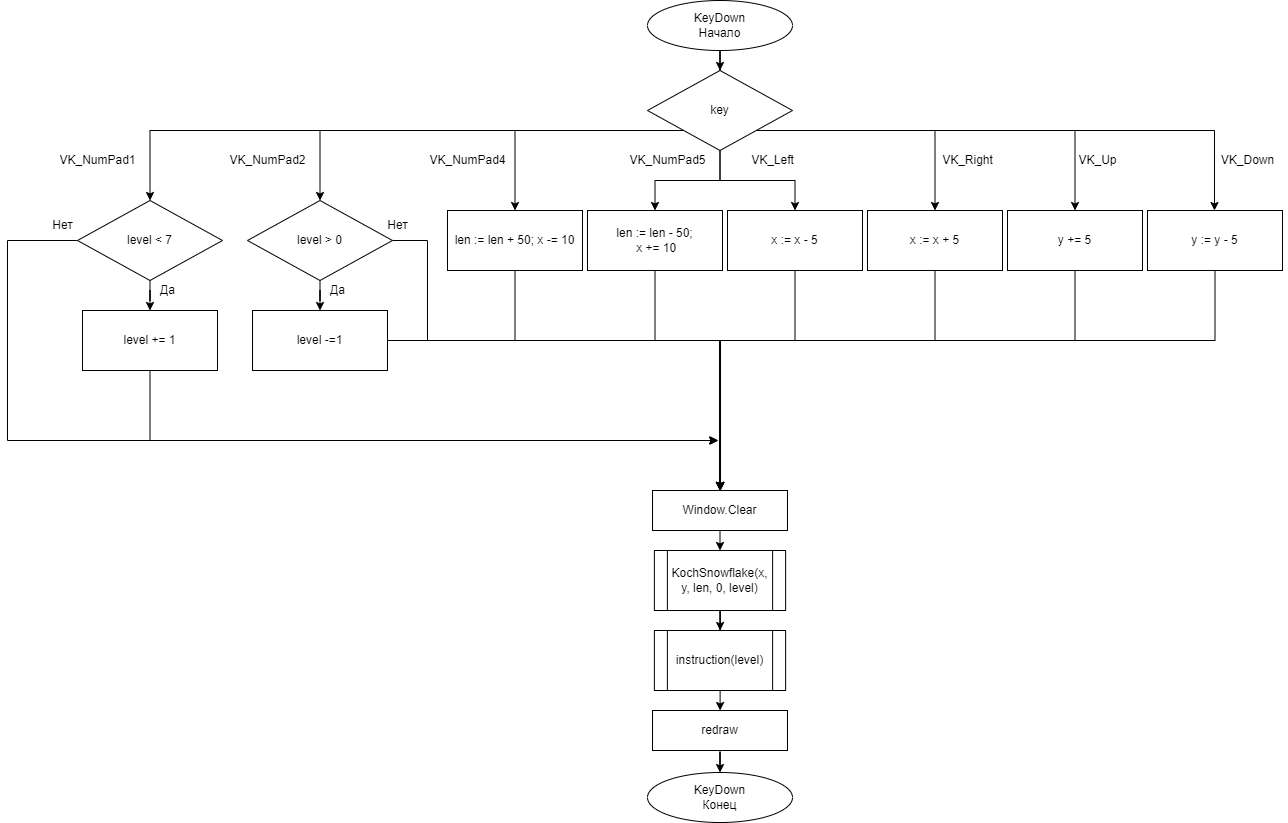


Рисунок 3 – алгоритм процедуры KeyDown

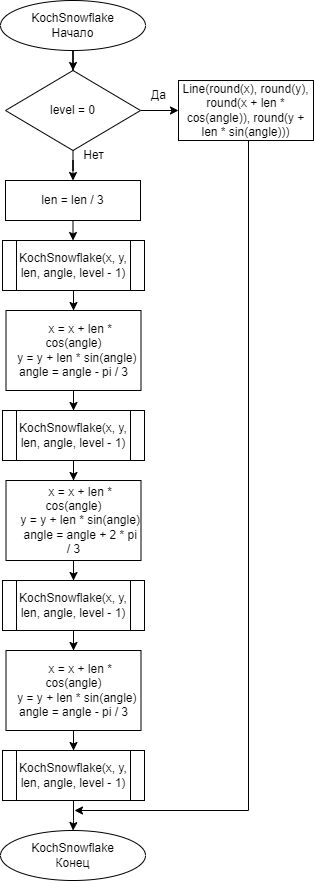


Рисунок 4 – алгоритм процедуры KochSnowflake

**Кода программы:**

**program** DKR\_KochSnowFlake;

**uses** recursia, GraphABC;

**var** x,y,level,len:integer;

**procedure** instruction(m:integer);

**begin**

textout(10,100,'Left - влево');

textout(10,120,'Right - вправо');

textout(10,140,'Up - вверх');

textout(10,160,'Down - вниз');

textout(10,180,'num 4 - увеличить масштаб');

textout(10,200,'num 5 - уменьшить масштаб');

textout(10,220,'num 1 - увеличить глубину');

textout(10,240,'num 2 - уменьшить глубину');

textout(10,260, abs(level));

textout(20,260, ' - Глубина рекурсии');

**end**;

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

**begin**

**case** key **of**

VK\_Down: **begin** y := y - 5 **end**;

VK\_Up: **begin** y += 5 **end**;

VK\_Right: **begin** x := x + 5 **end**;

VK\_Left: **begin** x := x - 5 **end**;

VK\_NumPad5: **begin** len := len - 50; x += 10 **end**;

VK\_NumPad4: **begin** len := len + 50; x -= 10 **end**;

VK\_NumPad2: **if** level>0 **then** level -= 1;

VK\_NumPad1:**if** level<7 **then** level += 1;

**end**;

Window.Clear;

KochSnowflake(x, y, len, 0, level);

instruction(level);

redraw;

**end**;

**begin**

LockDrawing;

x := 100;

y := 300;

len:=700;

level:= 0;

KochSnowflake(x, y, len, 0, level);

instruction(level);

redraw;

OnkeyDown := keydown;

**end**.

**unit** recursia;

**uses** GraphABC;

**procedure** KochSnowflake(x, y, len, angle: real; level: integer); // параметризация

**begin**

**if** level = 0 **then** // база рекурсии

**begin**

Line(round(x), round(y), round(x + len \* cos(angle)), round(y + len \* sin(angle)));

**end**

**else**

**begin**

len := len / 3;

KochSnowflake(x, y, len, angle, level - 1); // декомпозиция

x := x + len \* cos(angle);

y := y + len \* sin(angle);

angle := angle - pi / 3;

KochSnowflake(x, y, len, angle, level - 1); // декомпозиция

x := x + len \* cos(angle);

y := y + len \* sin(angle);

angle := angle + 2 \* pi / 3;

KochSnowflake(x, y, len, angle, level - 1); // декомпозиция

x := x + len \* cos(angle);

y := y + len \* sin(angle);

angle := angle - pi / 3;

KochSnowflake(x, y, len, angle, level - 1); // декомпозиция

**end**;

**end**;

**end**.

Результаты выполнения задания.

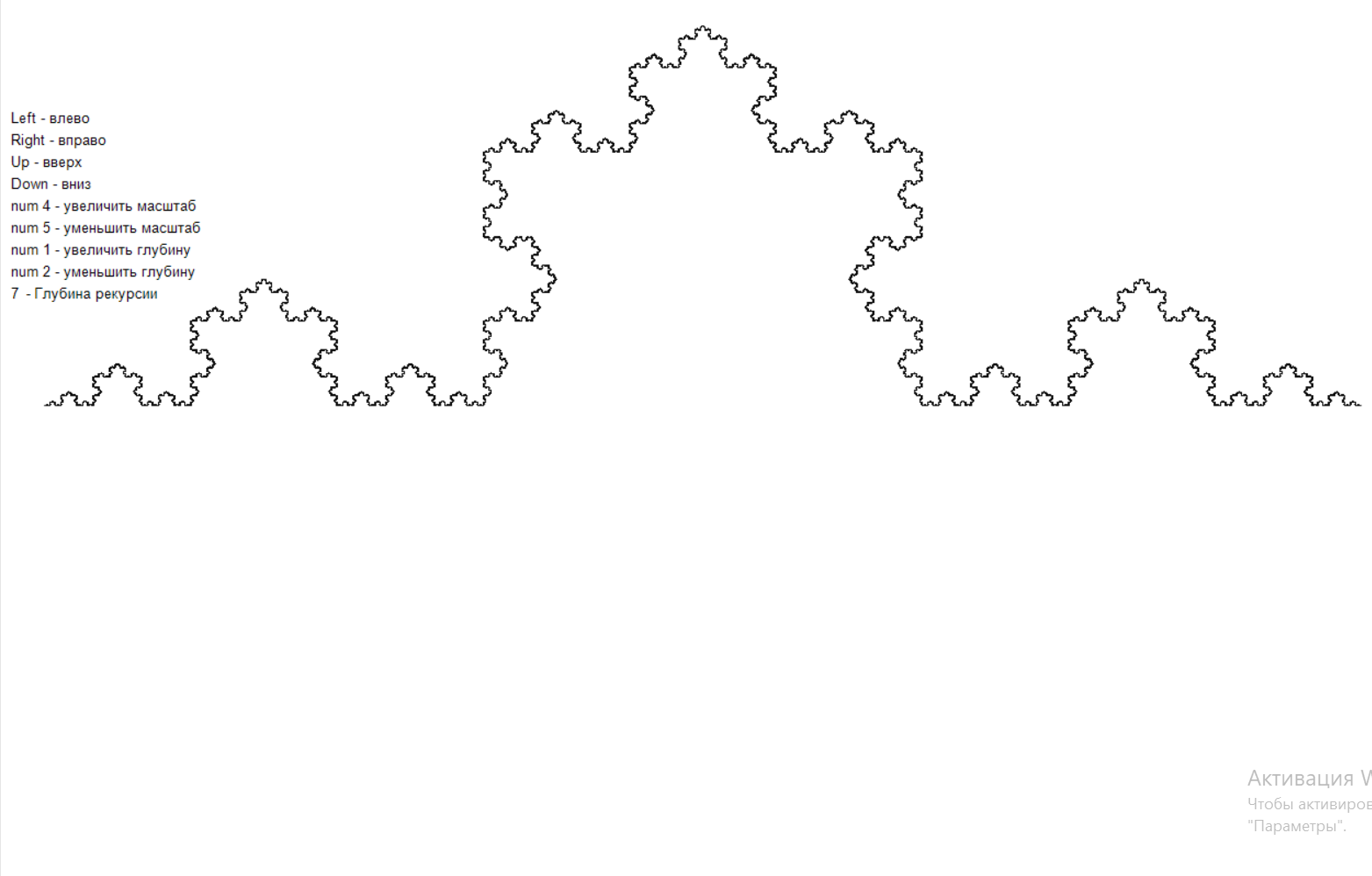


Рисунок 5 – результат выполнения задания

**Вывод:** в процессе выполнения задания проблем не возникло, был реализован фрактал «Снежинка Коха» с возможностью изменения положения и масштабированием. Так же в ходе работы рекурсивный алгоритм был вынесен в отдельный модуль, что позволит использовать эту рекурсию в других программах. Были получены навыки реализации рекурсивных алгоритмов и реализации фрактальной графики на языке программирования Pascal.