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-204-52-00

Вейс Роман Михайлович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. Цель работы

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

2. Формулировка задания.

Вариант 5.

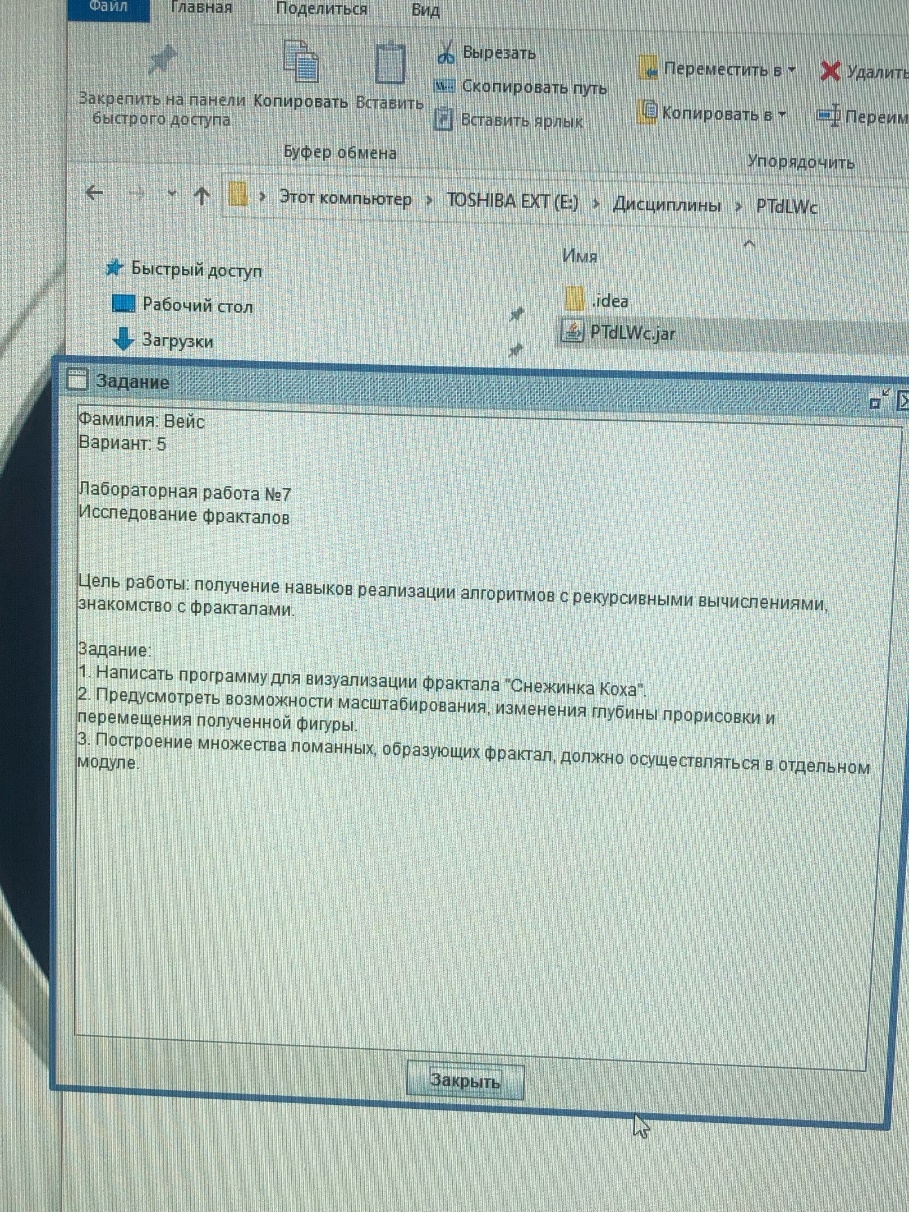


Рисунок 1 – Вариант работы.

3. Описание алгоритма.

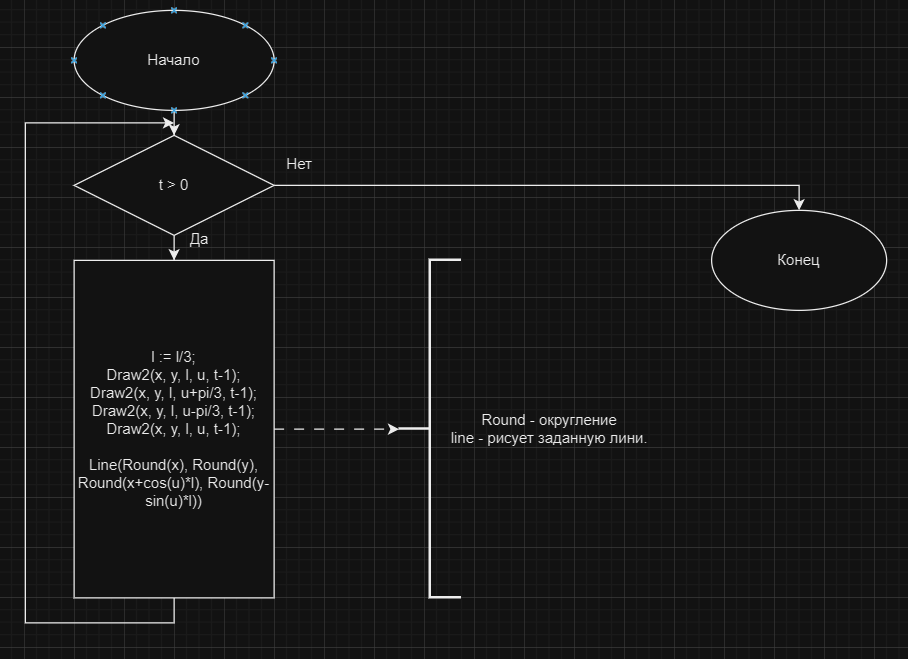
Данный алгоритм предназначен для создания фрактала «Снежинка Коха».

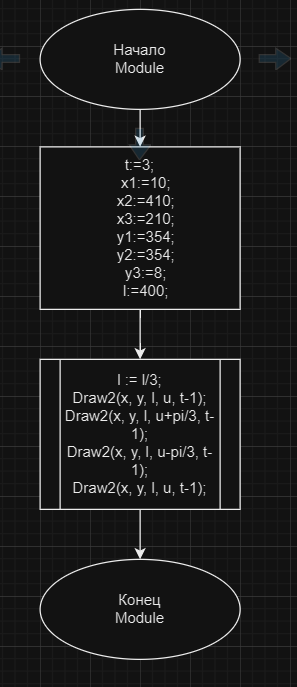
Изначально определяются переменные x1, x2, x3, y1, y2, y3, l для определения начальных координат и глубины фрактала.

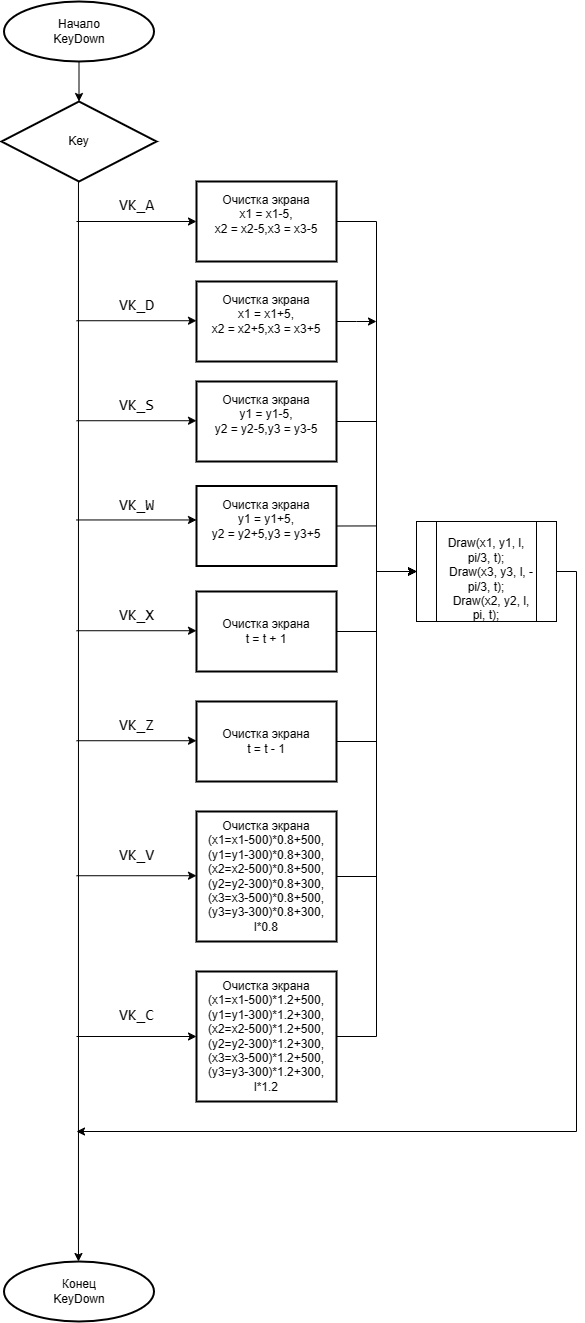
На каждой итерации алгоритм делит стороны треугольника на 3 части и затем повторяет процесс для каждой из получившихся частей, пока не достигнет заданной глубины.

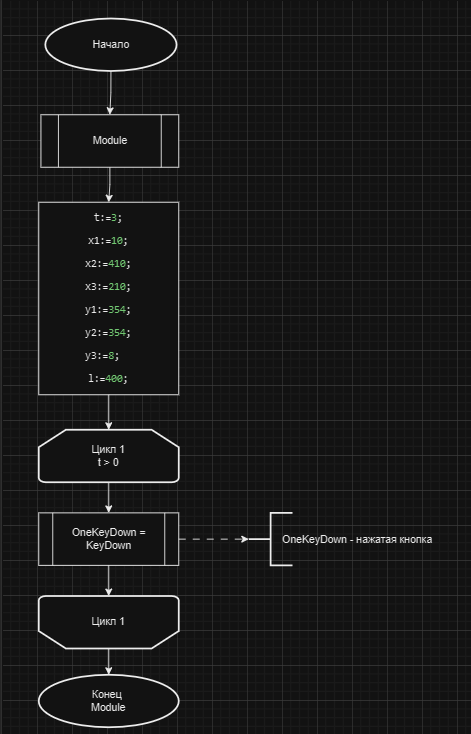
С помощью процедуры KeyDown, которая реагирует на нажатие клавиш, и в зависимости от нажатой клавиши (A, D, W, S, Z, X, C, V) изменяются координаты фрактала и глубина рекурсии. После каждого изменения происходит очистка окна и перерисовка фрактала с новыми параметрами.

4. Схема.









5. Код программы.

Задание 1.

Модуль.

**Unit** Kox;

**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**;

**begin**

SetWindowSize(1000,800);

**end**.

Основной алгоритм.

**uses**

GraphABC, Kox;

**var** t: integer;

x1, x2, x3, y1, y2 ,y3, l: real;

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

**begin**

**case** Key **of**

VK\_A:(x1,x2,x3):=(x1-5,x2-5,x3-5);

VK\_D:(x1,x2,x3):=(x1+5,x2+5,x3+5);

VK\_W:(y1,y2,y3):=(y1-5,y2-5,y3-5);

VK\_S:(y1,y2,y3):=(y1+5,y2+5,y3+5);

VK\_X:t:=t+1;

VK\_Z:t:=t-1;

VK\_V:(x1,y1,x2,y2,x3,y3,l):=((x1-500)\*0.8+500,(y1-300)\*0.8+300,(x2-500)\*0.8+500,(y2-300)\*0.8+300,(x3-500)\*0.8+500,(y3-300)\*0.8+300,l\*0.8);

VK\_C:(x1,y1,x2,y2,x3,y3,l):=((x1-500)\*1.2+500,(y1-300)\*1.2+300,(x2-500)\*1.2+500,(y2-300)\*1.2+300,(x3-500)\*1.2+500,(y3-300)\*1.2+300,l\*1.2);

**end**;

window.Clear;

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

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

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

**end**;

**begin**

t:=3;

x1:=10;

x2:=410;

x3:=210;

y1:=354;

y2:=354;

y3:=8;

l:=400;

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

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

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

OnKeyDown := KeyDown;

**end**.

6. Результат выполнения программы.

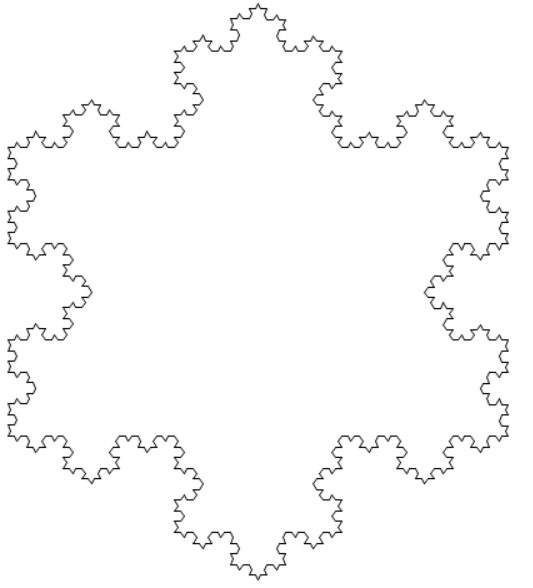


Рисунок 2

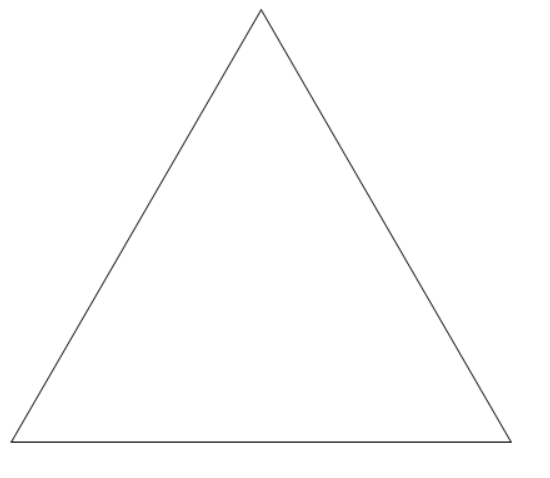


Рисунок 3

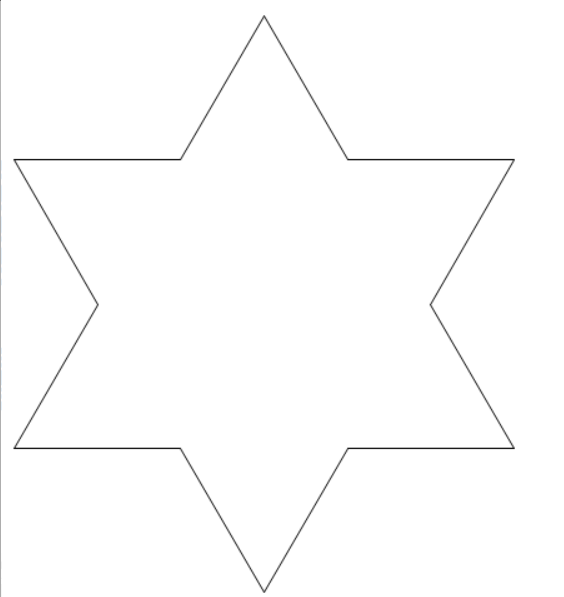
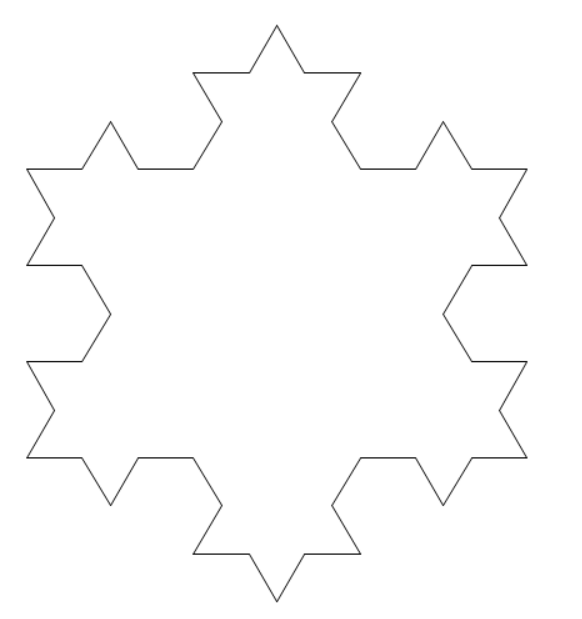


Рисунок 4

  
Рисунок 5

7. Вывод.

Выполняя домашнюю контрольную работу, мы смогли достичь поставленной цели: узнали, что такое фракталы и получили опыт работы с рекурсивными алгоритмами.

В ходе работы над задание мы изучили принципы работы с модулями.