Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №6**

**«ИССЛЕДОВАНИЕ ФРАКТАЛОВ»**

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

Выполнил: студент учебной группы

ИСПк-202-52-00

Колотова Алина

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**Задание 1.** Написать программу для визуализации фрактала "Кривая Леви".

**Задание 2.** Предусмотреть возможности масштабирования, изменения глубины прорисовки и перемещения полученной фигуры.

**Задание 3.** Построение множества ломанных, образующих фрактал, должно осуществляться в отдельном модуле.

# **Описание алгоритма**

Алгоритм работы LeviCurveModule:

Процедура DrawLevi

1. Базовый случай (level = 0): Если уровень рекурсии равен 0, процедура просто рисует прямую линию между начальной и конечной точками, используя функцию «Line» из модуля «GraphWPF».
2. Рекурсивный случай (level > 0): Если уровень рекурсии больше 0, процедура выполняет следующие действия: вычисление средней точки: Средняя точка отрезка линии вычисляется по формуле: x = (x0 + x1) / 2 + (y1 - y0) / 2, y = (y0 + y1) / 2 - (x1 - x0) / 2. Дважды вызывается процедура DrawLevi. Первый вызов рисует левую половину кривой: DrawLevi(level - 1, x0, y0, x, y), второй вызов рисует правую половину кривой: `DrawLevi(level - 1, x, y, x1, y1).

Алгоритм работы Main:

Процедура keyDown

1. Проверяется, находится ли программа в режиме рисования («painting»). Если нет, то обрабатываются нажатия клавиш:

key.Add или key.OemPlus: увеличивается level.

key.Subtract или key.OemMinus: уменьшается level, если он больше 1.

key.W, key.S, key.A, key.D: сдвигаются начальная и конечная точки (x0, y0) и (x1, y1) в соответствующих направлениях (вверх, вниз, влево, вправо).

key.Q или key.E: увеличивается или уменьшается масштаб.

1. Если была нажата любая другая клавиша, программа выводит сообщение о нажатой клавише и завершается.
2. Очищается окно рисования.
3. Вызывается процедура DrawLevi(level, x0, y0, x1, y1) из модуля LeviCurveModule для рисования кривой Леви.
4. Флаг painting` сбрасывается в значение false.
5. Обновляется заголовок окна с информацией о текущем уровне рекурсии и доступных командах управления.

Основное тело программы

Вызывается процедура DrawLevi(level, x0, y0, x1, y1) для рисования начальной кривой Леви.

1. **Схема алгоритма с комментариями**

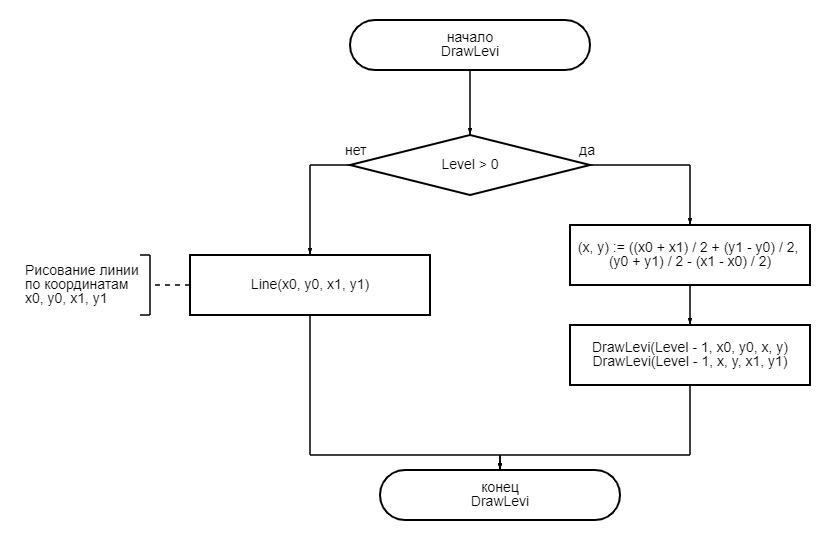
****

Рисунок 1 – Процедура DrawLevi модуля LeviCurveModule

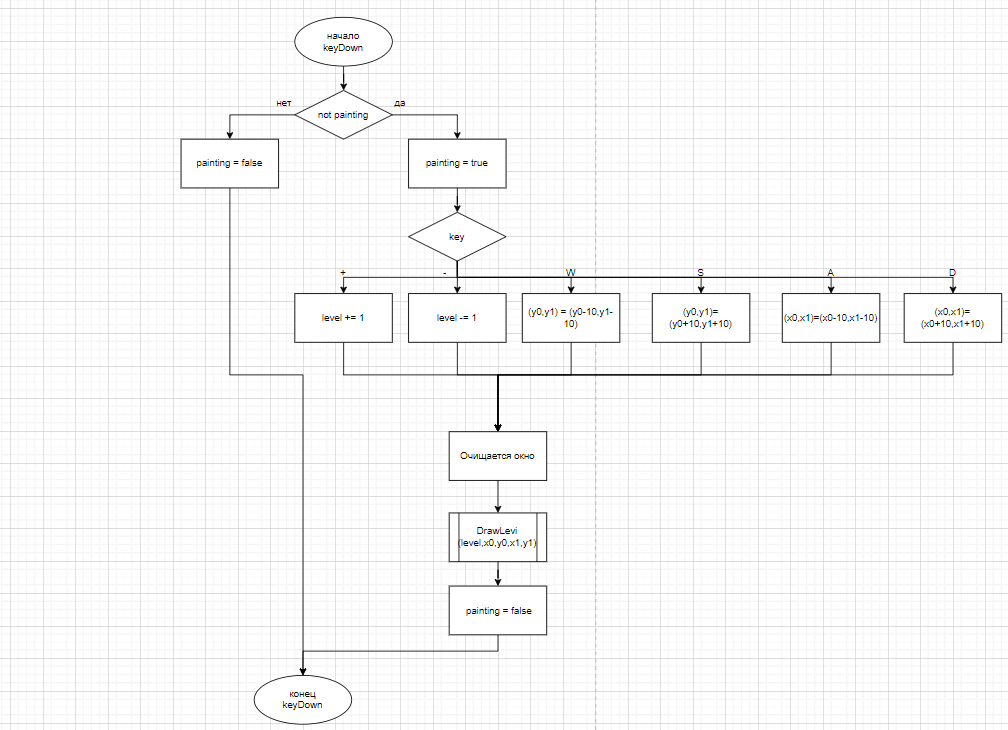


Рисунок 2 – Процедура KeyDown модуля Main

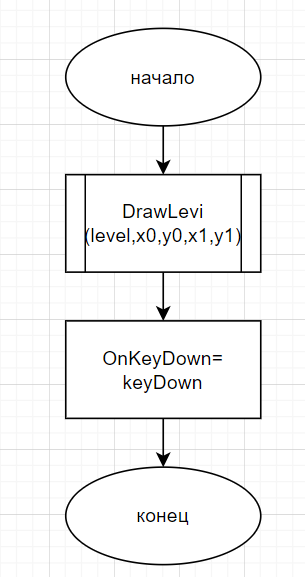


Рисунок 3 – Основное тело программы Main

1. **Код программы**

**unit** LeviCurveModule;

**interface**

**uses** GraphWPF;

**procedure** DrawLevi(Level: Integer; x0, y0, x1, y1: Real);

**implementation**

**procedure** DrawLevi(Level: Integer; x0, y0, x1, y1: Real);

**begin**

**if** Level > 0 **then**

**begin**

**var**

(x, y) := ((x0 + x1) / 2 + (y1 - y0) / 2, (y0 + y1) / 2 - (x1 - x0) / 2);

DrawLevi(Level - 1, x0, y0, x, y);

DrawLevi(Level - 1, x, y, x1, y1);

**end**

**else**

Line(x0, y0, x1, y1);

**end**;

**end**.

**program** Main;

**uses** GraphWPF, LeviCurveModule;

**var**

painting := false;

level := 1;

(x0,y0):=(Window.Width/3,Window.Height-Window.Height/3);

(x1,y1):=(Window.Width-Window.Width/3,Window.Height-Window.Height/3);

**procedure** keyDown(k:key);

**begin**

**if not** painting **then begin**

painting := true;

**case** k **of**

key.Add,key.OemPlus: level += 1;

key.Subtract,key.OemMinus: **if** level > 1 **then** level -= 1;

key.W : (y0,y1):=(y0-10,y1-10);

key.S : (y0,y1):=(y0+10,y1+10);

key.A : (x0,x1):=(x0-10,x1-10);

key.D : (x0,x1):=(x0+10,x1+10);

key.Q : (x0,y0,x1,y1):=((x0-window.Center.X)\*0.9+window.Center.x,(y0-window.Center.y)\*0.9+window.Center.y,(x1-window.Center.X)\*0.9+window.Center.x,(y0-window.Center.y)\*0.9+window.Center.y);

key.E : (x0,y0,x1,y1):=((x0-window.Center.X)\*1.1+window.Center.x,(y0-window.Center.y)\*1.1+window.Center.y,(x1-window.Center.X)\*1.1+window.Center.x,(y0-window.Center.y)\*1.1+window.Center.y);

**else**

window.Caption := $'key = {k}';

painting := false;

**exit**;

**end**;

window.Clear;

DrawLevi(level,x0,y0,x1,y1);

painting := false;

window.Caption:='Кривая Леви ' + level + ' +/- - глубина, WASD - сдвинуть, Q/E - дальше/ближе';

**end**;

**end**;

**begin**

DrawLevi(level,x0,y0,x1,y1);

OnKeyDown:=keyDown;

**end**.

1. **Результат выполнения программы**

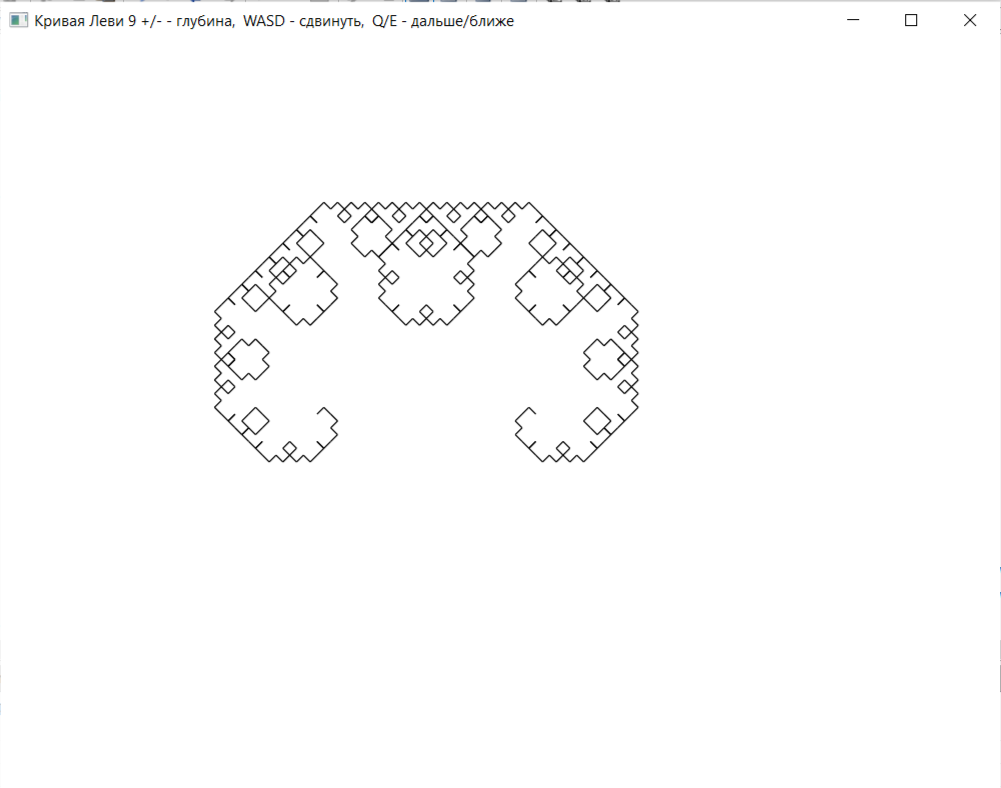
****

Рисунок 4 – Результат работы программы

1. **Вывод**

В ходе выполнения данной контрольной работы были получены навыки реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.

Фрактал — естественный объект или математическая модель объекта, обладающий свойством самоподобия, то есть в точности или приближённо совпадающий со своей частью.

Чтобы построить кривую Леви, нужно:

* Взять половину квадрата.
* Каждую сторону заменить таким же фрагментом.
* Повторять эту операцию.