Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №1**

**«Исследование фракталов»**

**ПО ДИСЦИПЛИНЕ «МДК.05.02»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Яковлев А.С.

Преподаватель:

Сергеева Е.Г.

Киров

2024

**Цель работы:**

Получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.

**Формулировка задания:**

* Написать программу для визуализации фрактала «Кривая Леви».
* Предусмотреть возможность масштабирования, изменения глубины прорисовки и перемещения полученной фигуры.
* Построение множества ломанных, образующих фрактал, должно осуществляться в отдельном модуле.

**Описание алгоритма и ответы на вопросы:**

Алгоритм, реализованный в модуле r, строит **кривую Леви** — фрактал, предложенный французским математиком П. Леви.

Чтобы получить кривую Леви, нужно взять половину квадрата, а затем каждую сторону заменить таким же фрагментом. Повторяя эту операцию, в пределе получится кривая Леви, напоминающую корону. В основной программе реализована работа кнопок: Q, W, E, R, T для изменения глубины, стрелки для перемещения фигуры, «+» и «-» для изменения масштаба.

**Схемы алгоритмов:**

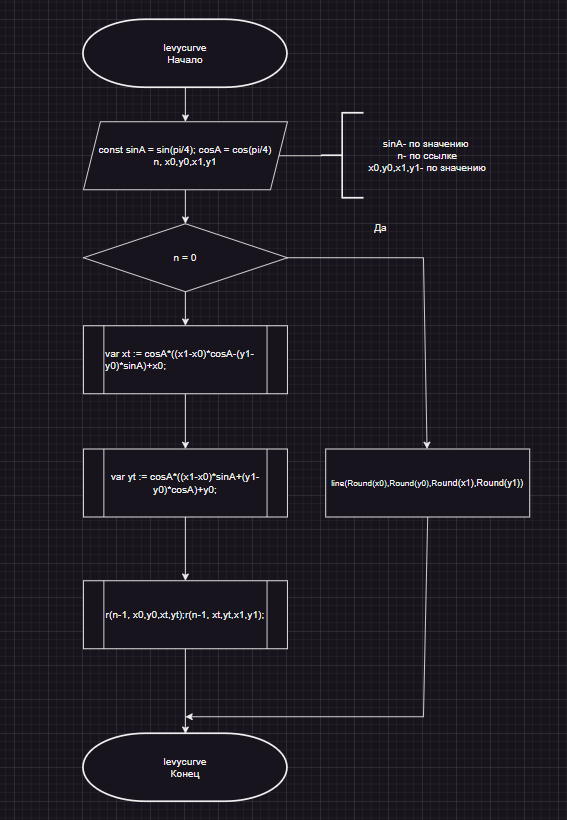


Рисунок 1 – схема алгоритма процедуры r в модуле LevyCurve

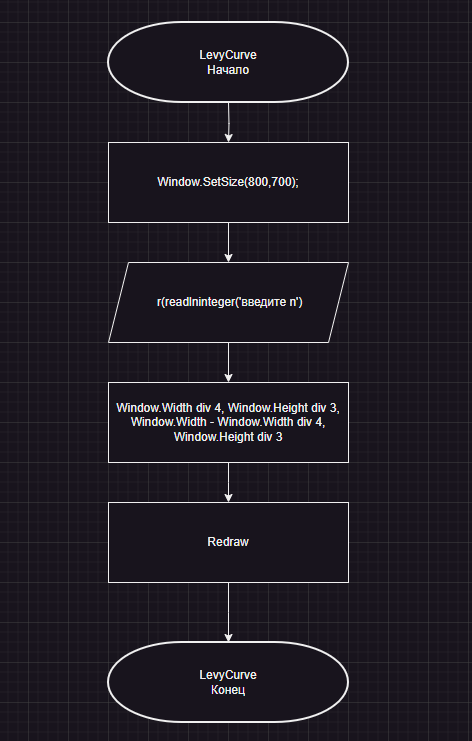


Рисунок 2 – схема алгоритма процедуры levycurve основной программы

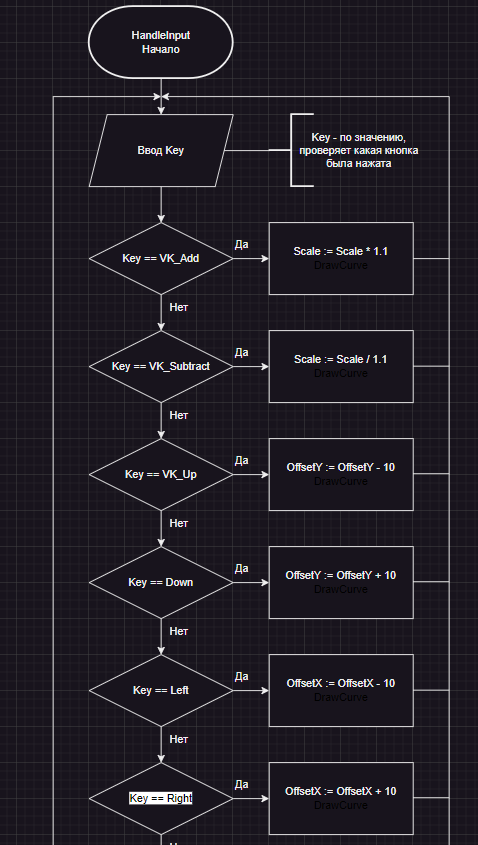


Рисунок 3 – схема алгоритма процедуры HandleInput

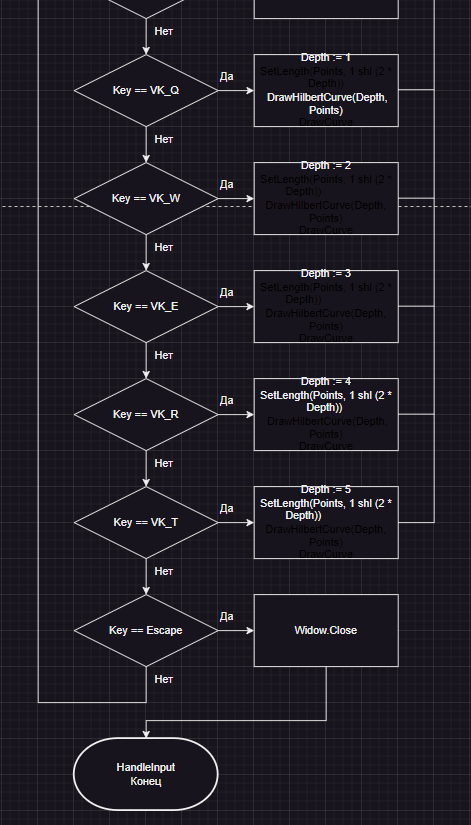


Рисунок 3.1 – схема алгоритма процедуры HandleInput

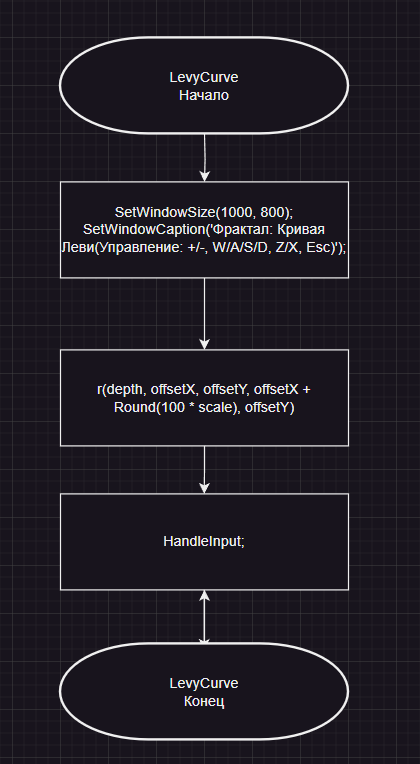


Рисунок 5 – схема алгоритма основной программы

**Код программы:**

**uses** GraphABC;

**const** sinA = sin(pi/4); cosA = cos(pi/4);

**procedure** r(n : integer; x0,y0,x1,y1 : real);

**begin**

**if** n = 0 **then** line(Round(x0),Round(y0),Round(x1),Round(y1))

**else**

**begin**

**var** xt := cosA\*((x1-x0)\*cosA-(y1-y0)\*sinA)+x0;

**var** yt := cosA\*((x1-x0)\*sinA+(y1-y0)\*cosA)+y0;

r(n-1, x0,y0,xt,yt);r(n-1, xt,yt,x1,y1);

**end**;

**end**;

**begin**

Window.SetSize(800,700);

r(readlninteger('введите n'), Window.Width **div** 4, Window.Height **div** 3, Window.Width - Window.Width **div** 4, Window.Height **div** 3);

Redraw;

**end**.

**uses** CRT, GraphABC, Levy;

**var**

scale: Real = 1.0; // масштаб

depth: Integer = 6; // глубина прорисовки

offsetX: Integer = 1;

offsetY: Integer = 1; // смещение фигуры

**procedure** HandleInput;

**var**

ch: Char;

**begin**

**repeat**

ch := ReadKey;

**case** ch **of**

#43: // Клавиша "+"

**begin**

depth := depth + 1; // увеличиваем глубину

r(depth, offsetX, offsetY, offsetX + Round(100 \* scale), offsetY); // обновляем отрисовку

ClearWindow;

**end**;

#45: // клавиша "-"

**begin**

**if** depth > 0 **then**

depth := depth - 1; // уменьшаем глубину

r(depth, offsetX, offsetY, offsetX + Round(100 \* scale), offsetY); // обновляем отрисовку

**end**;

#119, #87: // клавиши "W" или "w"

**begin**

ClearWindow;

offsetY := offsetY - 10; // перемещение вверх

r(depth, offsetX, offsetY, offsetX + Round(100 \* scale), offsetY); // обновляем отрисовку

**end**;

#115, #83: // клавиши "S" или "s"

**begin**

ClearWindow;

offsetY := offsetY + 10; // перемещение вниз

r(depth, offsetX, offsetY, offsetX + Round(100 \* scale), offsetY); // обновляем отрисовку

**end**;

#97, #65: // клавиши "A" или "a"

**begin**

ClearWindow;

offsetX := offsetX - 10; // перемещение влево

r(depth, offsetX, offsetY, offsetX + Round(100 \* scale), offsetY); // обновляем отрисовку

**end**;

#100, #68: // клавиши "D" или "d"

**begin**

ClearWindow;

offsetX := offsetX + 10; // перемещение вправо

r(depth, offsetX, offsetY, offsetX + Round(100 \* scale), offsetY); // обновляем отрисовку

**end**;

#122, #90: // клавиши "Z" или "z"

**begin**

scale := scale \* 1.1; // Увеличение масштаба

r(depth, offsetX, offsetY, offsetX + Round(100 \* scale), offsetY); // обновляем отрисовку

**end**;

#120, #88: // клавиши "X" или "x"

**begin**

scale := scale / 1.1; // уменьшение масштаба

r(depth, offsetX, offsetY, offsetX + Round(100 \* scale), offsetY); // обновляем отрисовку

**end**;

**end**;

**until** ch = #27; // выход по нажатию Esc

**end**;

**begin**

SetWindowSize(1000, 800);

SetWindowCaption('Фрактал: Кривая Леви(Управление: +/-, W/A/S/D, Z/X, Esc)');

r(depth, offsetX, offsetY, offsetX + Round(100 \* scale), offsetY); // первоначальная отрисовка

HandleInput; // обработка ввода

**end**.

**Результат выполнения программы:**

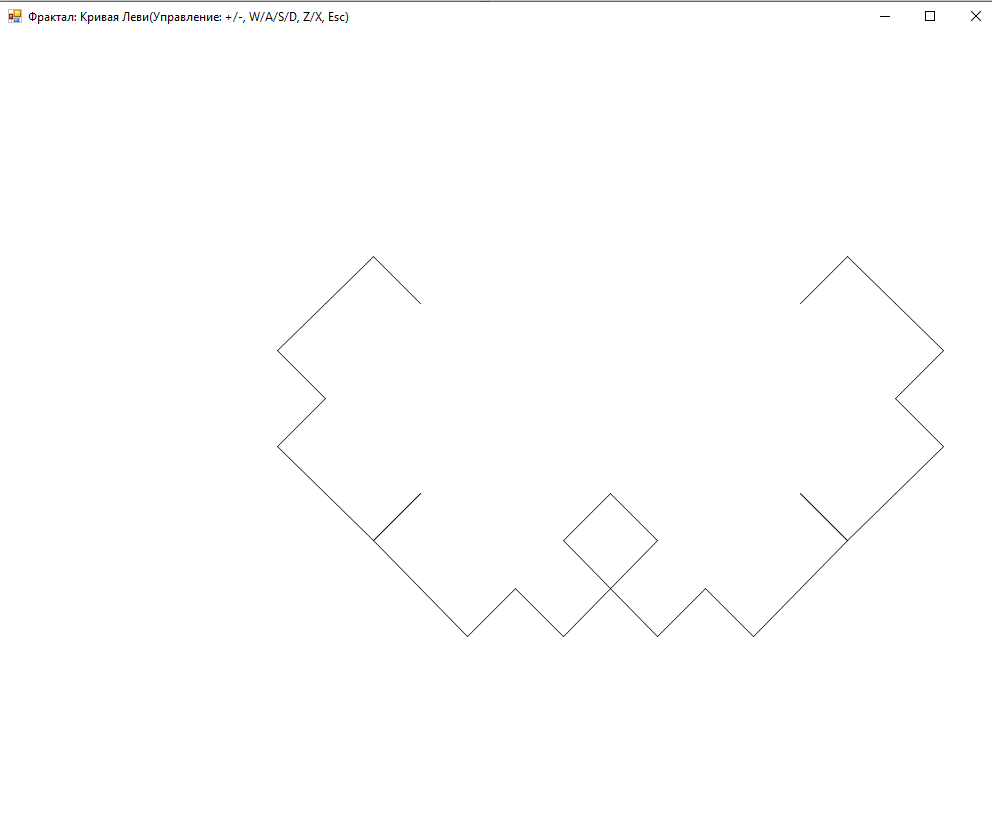
****

Рисунок 6 – результат выполнения работы одной из кнопок

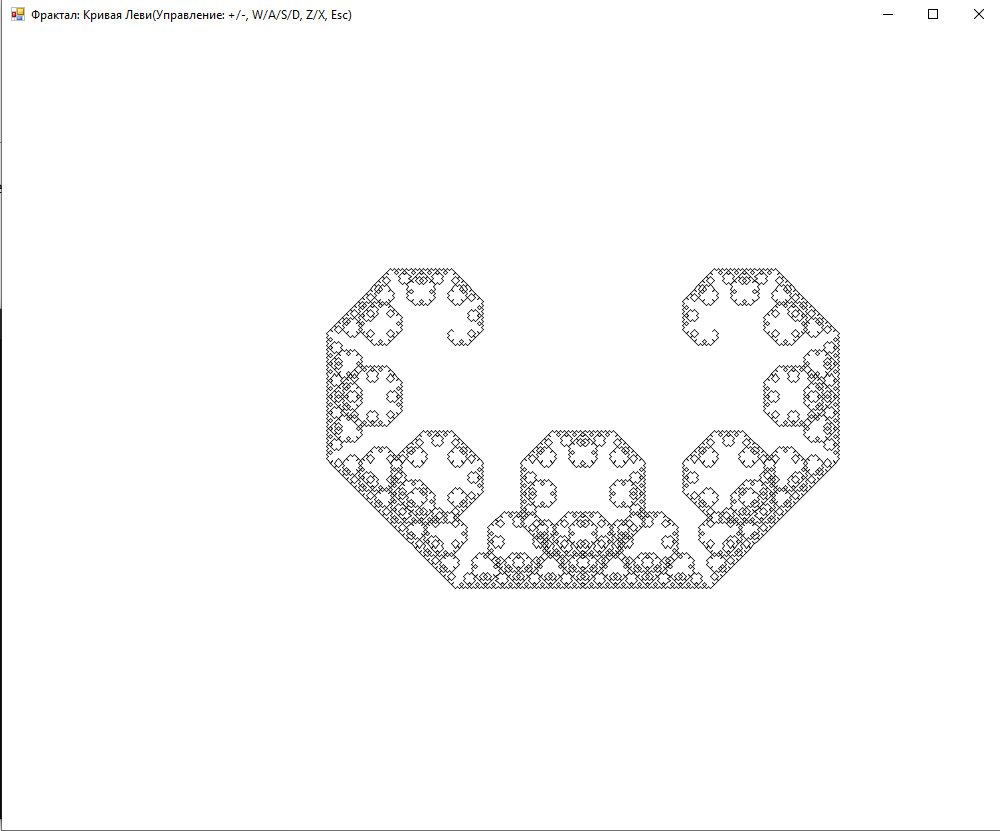


Рисунок 7 – результат выполнения одной из кнопок

**Вывод:**

Изучив правила построения кривой Леви, а также теоретический материал про рекурсивные алгоритмы, можно сделать следующий вывод: работа показалась довольно интересной, но сложной, также пришлось изучить информацию про модули и подключение работы клавиш. Фракталы довольно интересная тема для изучения и дальнейшего исследования.