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»**

**«МДК05.02 Разработка кода инофрмационных систем»**

Выполнил: студент учебной группы

ИСПк-205-52-00

Сальченко Егор Вадимович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

Цель работы

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

Задание:

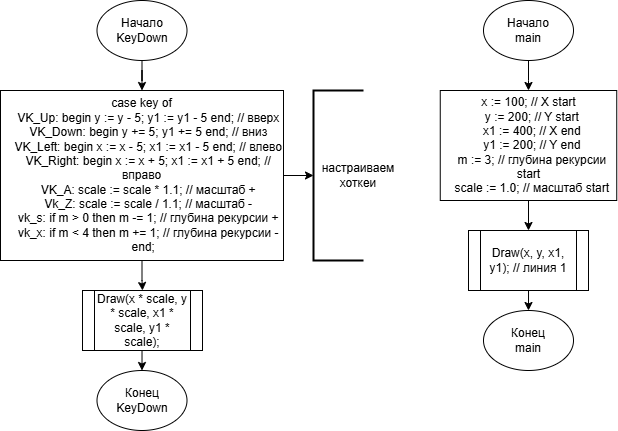
Формулировка задания (с вариантом)

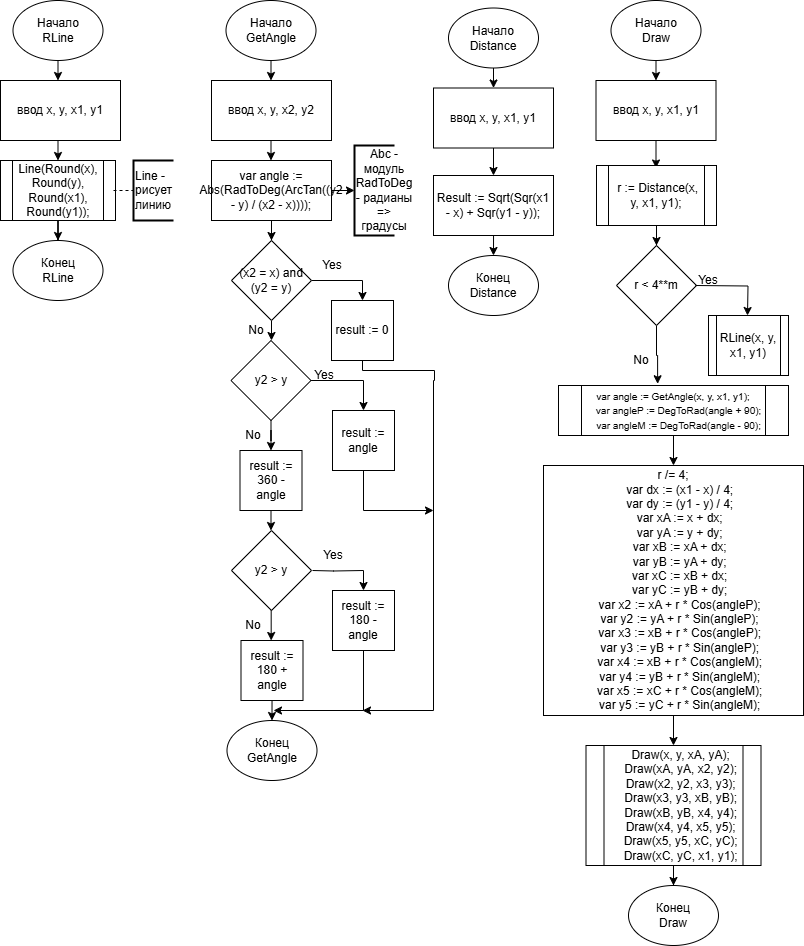
Вариант 15

﻿﻿ Задание:

1. Написать программу для визуализации фрактала "Кривая Минковского".
2. Предусмотреть возможности масштабирования, изменения глубины прорисовки и перемещения полученной фигуры.
3. Построение множества ломанных, образующих фрактал, должно осуществляться в отдельном модуле

**Схема алгоритма**

****



Код для программы в Pascal.

**Задание 1**

**Главный код**

uses GraphABC, fractals;

procedure KeyDown(key: integer); // обрабатываем кнопки

begin

case key of

VK\_Up: begin y := y - 5; y1 := y1 - 5 end; // вверх

VK\_Down: begin y += 5; y1 += 5 end; // вниз

VK\_Left: begin x := x - 5; x1 := x1 - 5 end; // влево

VK\_Right: begin x := x + 5; x1 := x1 + 5 end; // вправо

VK\_A: scale := scale \* 1.1; // масштаб +

Vk\_Z: scale := scale / 1.1; // масштаб -

vk\_s: if m > 0 then m -= 1; // глубинa рекурсии +

vk\_x: if m < 4 then m += 1; // глубинa рекурсии -

end;

Window.Clear;

Draw(x \* scale, y \* scale, x1 \* scale, y1 \* scale); // применить масштаб

Redraw;

end;

begin // ПАРАМЕТРИЗАЦИЯ

LockDrawing; // оптимизация отрисовки

x := 100; // X start

y := 200; // Y start

x1 := 400; // X end

y1 := 200; // Y end

m := 3; // глубина рекурсии start

scale := 1.0; // масштаб start

Draw(x, y, x1, y1); // линия 1

Redraw; // обновляем экран

OnKeyDown += KeyDown; // обработчик событий клавиатуры start

end.

unit fractals;

interface

uses GraphABC;

// Рисует линию от (x, y) до (x1, y1), округляя координаты до целых

procedure RLine(x, y, x1, y1: real);

// Возвращает угол между двумя точками (в градусах)

function GetAngle(x, y, x2, y2: real): real;

// Возвращает расстояние между двумя точками

function Distance(x, y, x1, y1: real): real;

// Основная рекурсивная процедура, рисующая фрактал

procedure Draw(x, y, x1, y1: real);

// Глобальные переменные

var

m: integer; // глубина рекурсии

scale: real; // масштаб изображения

x, y, x1, y1: integer; // координаты начальной линии

implementation

// Рисует линию с округлением координат

procedure RLine(x, y, x1, y1: real);

begin

Line(Round(x), Round(y), Round(x1), Round(y1));

end;

// Вычисляет угол между точками в градусах (0–360)

function GetAngle(x, y, x2, y2: real): real;

begin

var angle := Abs(RadToDeg(ArcTan((y2 - y) / (x2 - x))));

if (x2 = x) and (y2 = y) then

Result := 0

else if x2 > x then

if y2 > y then Result := angle else Result := 360 - angle

else if y2 > y then Result := 180 - angle else Result := 180 + angle;

end;

// Вычисляет расстояние между точками

function Distance(x, y, x1, y1: real): real;

begin

Result := Sqrt(Sqr(x1 - x) + Sqr(y1 - y));

end;

// Рекурсивная процедура рисования кривой Минковского

procedure Draw(x, y, x1, y1: real);

begin

var r := Distance(x, y, x1, y1); // Вычисляем расстояние между точками

// Базовый случай рекурсии: если отрезок слишком короткий, просто рисуем его

if r < 4\*\*m then

RLine(x, y, x1, y1) // БАЗОВЫЙ СЛУЧАЙ

else // ДЕКОМПОЗИЦИЯ

begin

var angle := GetAngle(x, y, x1, y1); // Угол направления отрезка

var angleP := DegToRad(angle + 90); // Угол для выступа вверх

var angleM := DegToRad(angle - 90); // Угол для выступа вниз

r /= 4; // делим длину отрезка на 4

// Вычисляем смещение по x и y для одной четверти отрезка

var dx := (x1 - x) / 4;

var dy := (y1 - y) / 4;

// Промежуточные точки деления отрезка на 4 части

var xA := x + dx;

var yA := y + dy;

var xB := xA + dx;

var yB := yA + dy;

var xC := xB + dx;

var yC := yB + dy;

// Выступы вверх (x2, x3) и вниз (x4, x5) от основной линии

var x2 := xA + r \* Cos(angleP);

var y2 := yA + r \* Sin(angleP);

var x3 := xB + r \* Cos(angleP);

var y3 := yB + r \* Sin(angleP);

var x4 := xB + r \* Cos(angleM);

var y4 := yB + r \* Sin(angleM);

var x5 := xC + r \* Cos(angleM);

var y5 := yC + r \* Sin(angleM);

// Рекурсивно рисуем 8 отрезков, заменяющих исходный: РЕКУРСИВНАЯ ПРОЦЕДУРА

Draw(x, y, xA, yA); // первая четверть

Draw(xA, yA, x2, y2); // вниз

Draw(x2, y2, x3, y3); // горизонталь внизу

Draw(x3, y3, xB, yB); // вверх

Draw(xB, yB, x4, y4); // вверх

Draw(x4, y4, x5, y5); // горизонталь вверху

Draw(x5, y5, xC, yC); // вниз

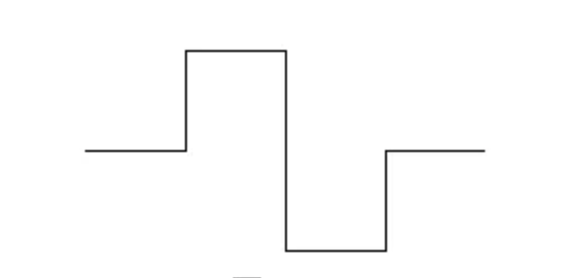
Draw(xC, yC, x1, y1); // последняя четверть

end;

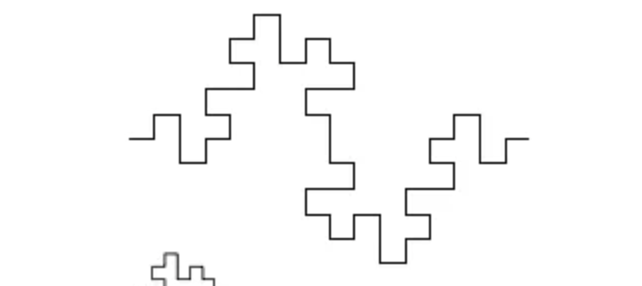
end;

end.

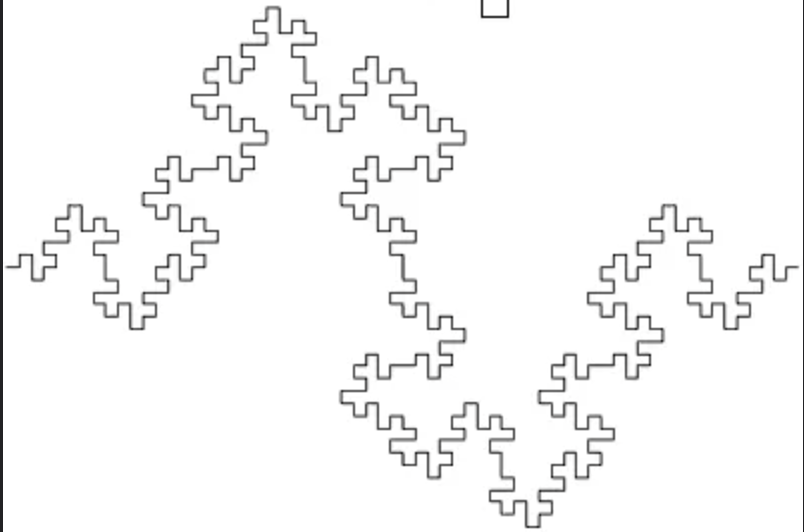
Результат программы













Вывод:

С помощью домашней контрольной работы №7 я научился строит фрактал "Кривая Минковского" с возможностью масштабирования, изменения глубины прорисовки и перемещения, разделяя построение фрактала и визуализацию на отдельные модули.