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

Титков Дмитрий Михайлович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. Цель работы: Получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.
2. Задание:
3. Написать программу для визуализации фрактала "Кривая Минковского".
4. Предусмотреть возможности масштабирования, изменения глубины прорисовки и перемещения полученной фигуры.
5. Построение множества ломанных, образующих фрактал, должно осуществляться в отдельно модуле.

Описание алгоритма

Была реализована рекурсивная функция для построения фрактала.

Фрактал (от лат. fractus — дроблёный, сломанный, разбитый) представляет собой сложную геометрическую фигуру с бесконечной самоподобностью, каждый фрагмент которой повторяется при изменении масштаба.

Фрактал "Кривая Минковского" строится следующим образом: он состоит из восьми отрезков одинаковой длины, соединённых по определённому принципу. Порядок соединения таков: отрезок вправо, отрезок вверх, два отрезка вниз, отрезок вправо, отрезок вверх и снова отрезок вправо. Длина каждого отрезка составляет четверть разности длины исходного отрезка, заданного его начальной и конечной точками.

Функция построения кривой вынесена в отдельный модуль, обозначенный в основной программе как М. Для графической визуализации использовалась библиотека GraphABC.

Кроме того, были созданы отдельная процедура, обрабатывающая действия фигуры при нажатии клавиш. Они реализованы с помощью case-меню и изменяют координаты, используемые функцией построения кривой, в зависимости от нажатых клавиш.

В основном теле программы обработчик событий KeyDown привязывается к соответствующим процедурам, управляющим действиями фигуры при нажатии клавиш.

1. Схема алгоритма с комментариями.

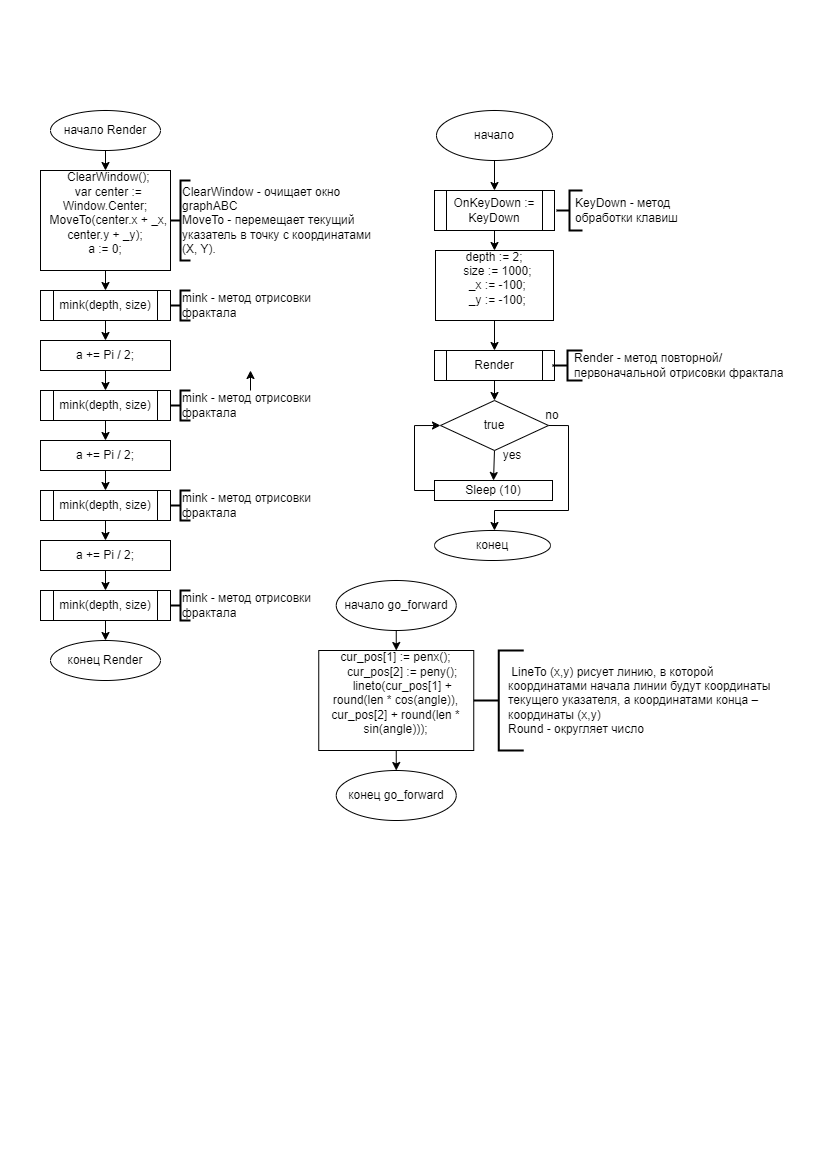


Рисунок 1 — Процедуры для построения и отрисовки фрактала и тело программы

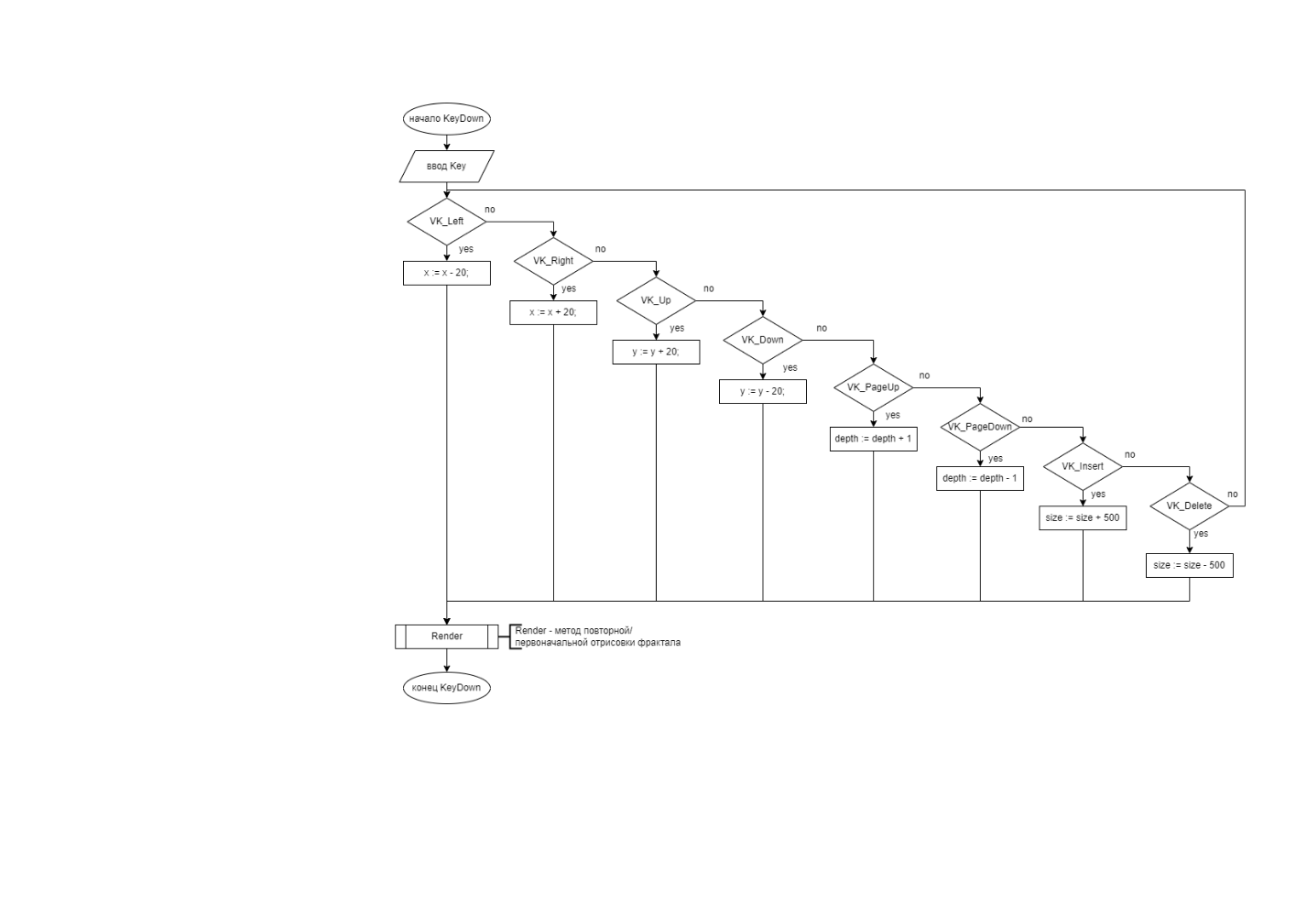


Рисунок 2 — Процедура для обработки клавиш

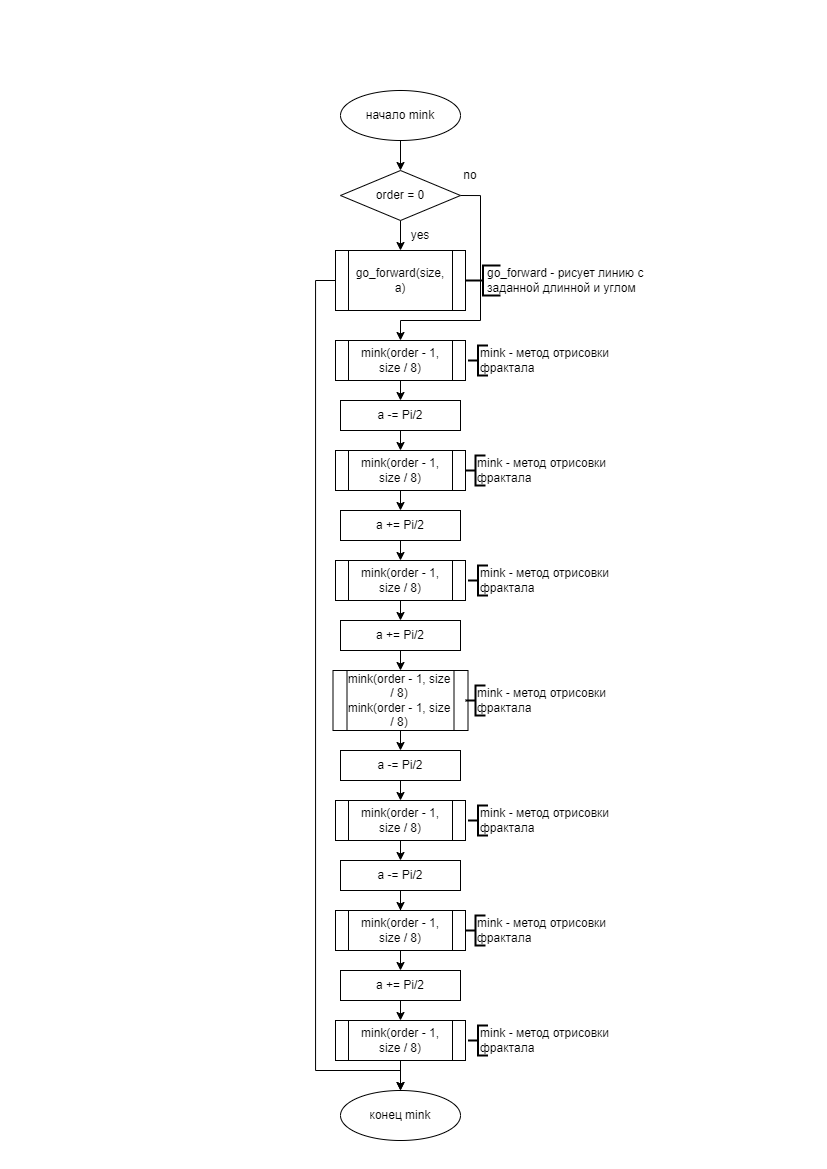


Рисунок 3 — Процедура для построения фрактала

Код программы

uses graphabc;

uses fractals;

var size: real ;

var depth: integer;

var \_x, \_y: integer;

procedure Render();

begin

ClearWindow();

var center := Window.Center;

MoveTo(center.x + \_x, center.y + \_y);

a := 0;

mink(depth, size);

a += Pi / 2;

mink(depth, size);

a += Pi / 2;

mink(depth, size);

a += Pi / 2;

mink(depth, size);

end;

procedure KeyDown(Key: integer);

begin

case Key of

VK\_Left: \_x := \_x - 20; //move right

VK\_Right: \_x := \_x + 20; //move left

VK\_Up: \_y := \_y - 20; //move up

VK\_Down: \_y := \_y + 20; //move down

VK\_PageUp: depth := depth + 1; //add recursive's depth

VK\_PageDown: depth := depth - 1;//remove recursive's depth

VK\_Insert: size := size + 500; //add size

VK\_Delete: size := size - 500; //remove size

end;

Render(); //call drawing after coordinates change

end;

begin

//depth := readinteger('input recursion s depth from 1 to 4: ');

//size := readreal('input fractal s size from 500 to 3000: ');

//\_x := readinteger('input x position: ');

//\_y := readinteger('input y position: ');

OnKeyDown := KeyDown; //install key handler

depth := 2;

size := 1000;

\_x := -100;

\_y := -100;

Render(); //first render

while true do

begin

Sleep(10); //reducing processor's load

end;

end.

**Unit**;

unit fractals;

interface

uses GraphABC;

var

a: real; // global variable for the current angle

procedure go\_forward(len: real; angle: real); // drawing a line of a given length and angle

procedure mink(order: integer; size: real); // drawing the Minkowski curve

implementation

// drawing a line of a given length and angle

procedure go\_forward;

var

cur\_pos: array[1..2] of integer;

begin

cur\_pos[1] := penx();

cur\_pos[2] := peny();

lineto(cur\_pos[1] + round(len \* cos(angle)), cur\_pos[2] + round(len \* sin(angle)));

end;

// drawing the Minkowski curve

procedure mink;

begin

if order = 0 then

go\_forward(size, a)

else

begin

mink(order - 1, size / 8);

a -= Pi / 2;

mink(order - 1, size / 8);

a += Pi / 2;

mink(order - 1, size / 8);

a += Pi / 2;

mink(order - 1, size / 8);

mink(order - 1, size / 8);

a -= Pi / 2;

mink(order - 1, size / 8);

a -= Pi / 2;

mink(order - 1, size / 8);

a += Pi / 2;

mink(order - 1, size / 8);

end;

end;

end.

Результат выполнения программы

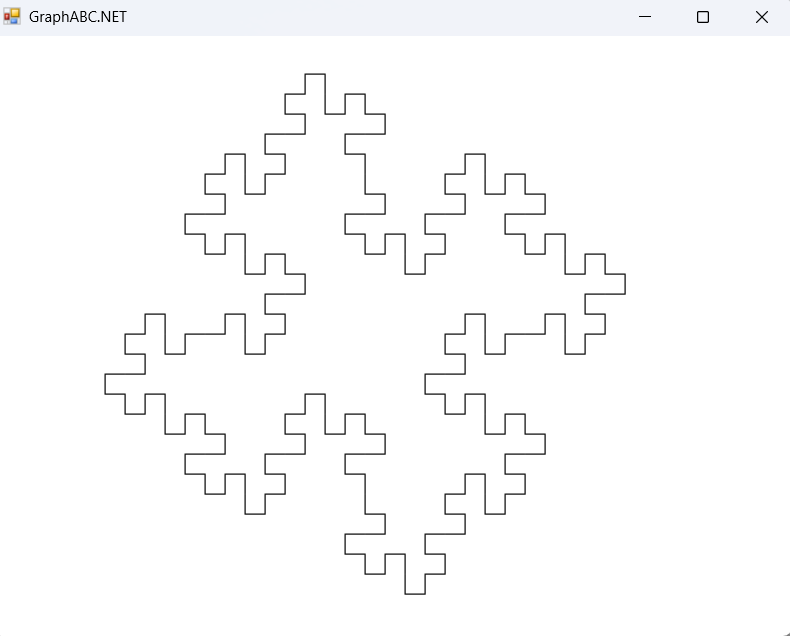


Рисунок 4 — Первоначальная отрисовка

Рисунок 5 — Изменение масштаба

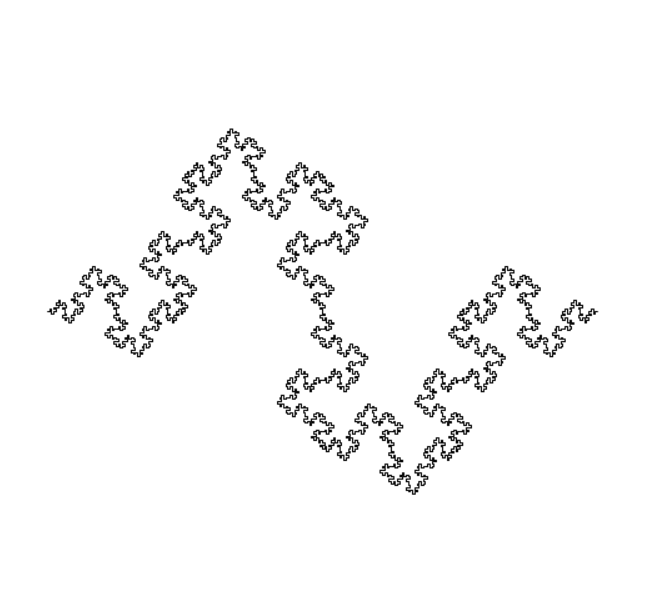


Рисунок 6 — Изменение глубины

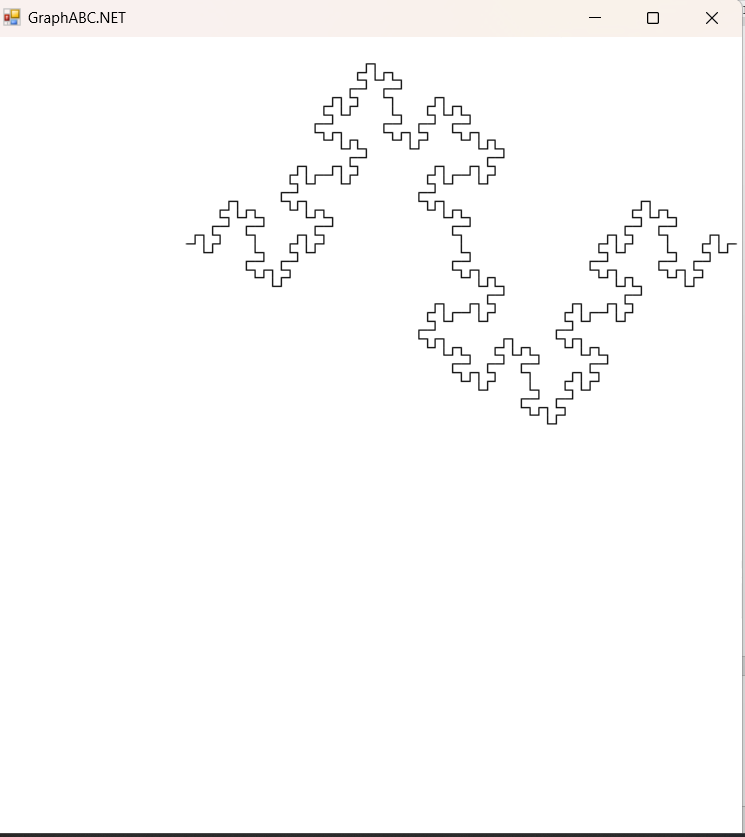


Рисунок 7 — изменение положения фигуры

Вывод

В ходе выполнения контрольной работы по исследованию фракталов были изучены их основные понятия, структуры и свойства.

Анализ фракталов способствует более глубокому пониманию таких концепций, как бесконечность и сложность, а также их практическому применению в различных областях науки и искусства. В рамках работы был подробно рассмотрен фрактал Минковского.

Результаты экспериментов подтверждают, что фракталы находят широкое применение в таких сферах, как компьютерная графика, медицина, биология и физика. Фрактальная геометрия также используется для моделирования и анализа природных явлений, включая облака, горные массивы и береговые линии.

Таким образом, исследование фракталов открывает новые перспективы для научных изысканий и практических приложений, демонстрируя, что за внешней простотой может скрываться бесконечная сложность.

Выводы работы подчеркивают значимость дальнейших исследований в этой области, способствующих развитию технологий и более глубокому пониманию окружающего мира.