**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»**

Выполнила: студент учебной группы

ИСПк-204-52-00

Панкратов Никита Владимирович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Цель работы**

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

**Формулировка задания**

Вариант 14

1. Написать программу для визуализации фрактала “Кривая Хартера-Хейтуэя”

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

3. Построение множества множества ломанных, образующих фрактал, изменения глубины прорисовки и перемещения полученной фигуры.

**Описание алгоритма**

Данный алгоритм рисует фрактал "Кривая дракона" с помощью графической библиотеки GraphABC и предоставлять возможность масштабировать фрактал.

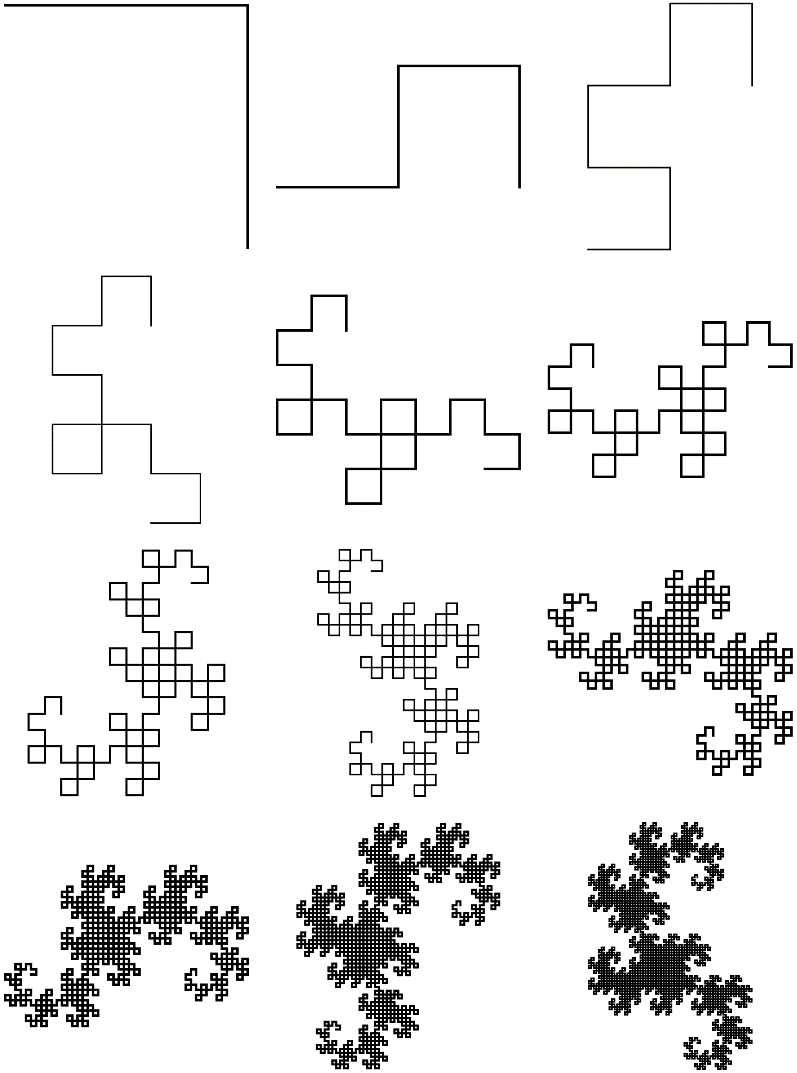
Фрактал дракона является одним из наиболее известных фракталов и может быть построен с использованием простых правил рекурсивного построения. Вот базовые шаги для построения фрактала дракона:

1. Начните с отрезка. Это будет первое "чешуйка" дракона.

2. Поверните отрезок на 90 градусов против часовой стрелки и разделите его на две части. Это будут две "чешуйки" дракона.

3. Повторите процесс для каждой новой "чешуйки" дракона: поверните ее на 90 градусов против часовой стрелки и разделите на две части.

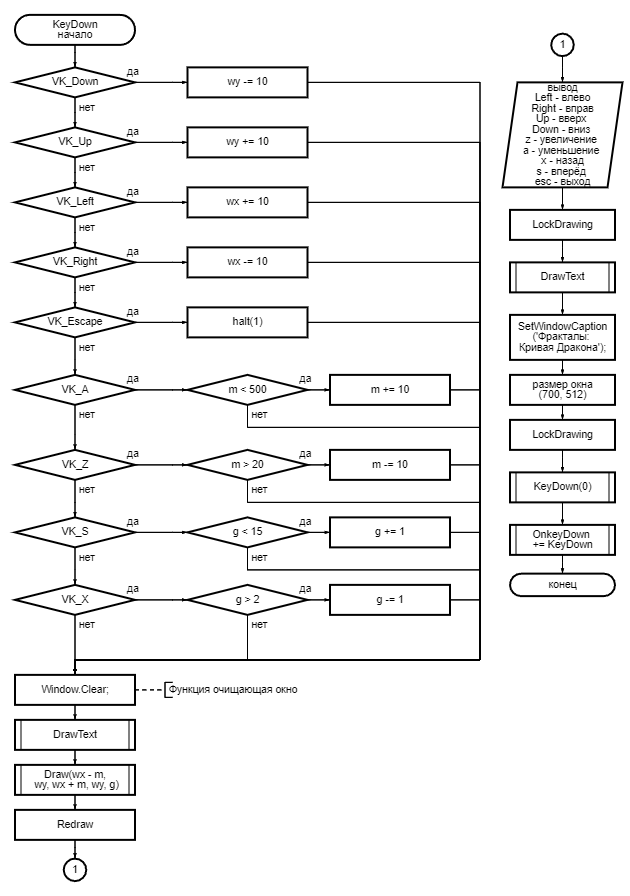
4. Продолжайте повторять шаги 2-3 для новых "чешуек" дракона, увеличивая их количество.



**Схема алгоритма с комментариями**



Рис. 1 «Схема алгоритма подпрограммы DrawText»

  
Рис. 2 «Схема алгоритма подпрограммы KeyDown»

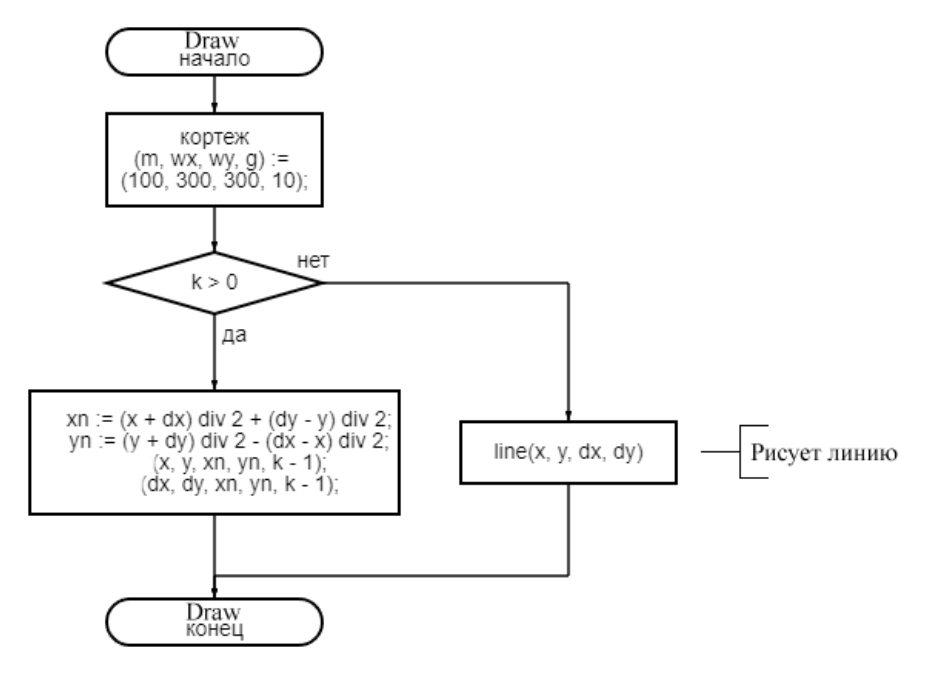


Рис. 3 «Схема алгоритма модуля rec»

**Код программы**

**uses** rec;

**uses** GraphABC;

// Процедура для вывода текста на экран

**procedure** DrawText;

**begin**

// Текстовые подсказки

TextOut(10, 100, 'Left - влево');

TextOut(10, 120, 'Right - вправо');

TextOut(10, 140, 'Up - вверх');

TextOut(10, 160, 'Down - вниз');

TextOut(10, 180, 'z - увеличение');

TextOut(10, 200, 'a - уменьшение');

TextOut(10, 220, 'x - назад');

TextOut(10, 240, 's - вперёд');

TextOut(10, 260, 'esc - выход');

**end**;

// Масштабирование

**Procedure** KeyDown(k: integer);

**begin**

**case** K **of**

VK\_Down: wy -= 10;

VK\_Up: wy += 10;

VK\_Left: wx += 10;

VK\_Right: wx -= 10;

VK\_Escape: halt(1);

VK\_A: **if** m < 500 **then** m += 10;

VK\_Z: **if** m > 20 **then** m -= 10;

VK\_S: **if** g < 15 **then** g += 1;

VK\_X: **if** g > 2 **then** g -= 1;

**end**;

Window.Clear;

DrawText;

Draw(wx - m, wy, wx + m, wy, g); // отрисовка фрактала

Redraw;

**end**;

**begin**

// Вывод подсказок на экран

writeln('Left - влево');

writeln('Right - вправо');

writeln('Up - вверх');

writeln('Down - вниз');

writeln('z - увеличение');

writeln('a - уменьшение');

writeln('x - назад');

writeln('s - вперёд');

writeln('esc - выход');

LockDrawing; // блокировка отрисовки

// Основной блок программы

**begin**

LockDrawing;

DrawText; // вывод текста

SetWindowCaption('Фракталы: Кривая Дракона');

SetWindowSize(700, 512);

LockDrawing;

KeyDown(0);

OnkeyDown += KeyDown;

**end**;

**end**.

**Модуль rec**

**unit** rec;

**uses** GraphABC;

// Параметризация

**var** (m, wx, wy, g) := (100, 300, 300, 10);

// База рекурсии и декомпозиция

**Procedure** Draw(x, y, dx, dy, k: integer);

**begin**

// База рекурсии

**If** k > 0 **then**

**begin**

**var** xn := (x + dx) **div** 2 + (dy - y) **div** 2; // декомпозиция

**var** yn := (y + dy) **div** 2 - (dx - x) **div** 2; // декомпозиция

Draw(x, y, xn, yn, k - 1); //рекурсия вызывает сама себя

Draw(dx, dy, xn, yn, k - 1); //рекурсия вызывает сама себя

**end**

**else**

line(x, y, dx, dy); // база рекурсии

**end**;

**begin**

**end**.

**Результат выполнения программы**

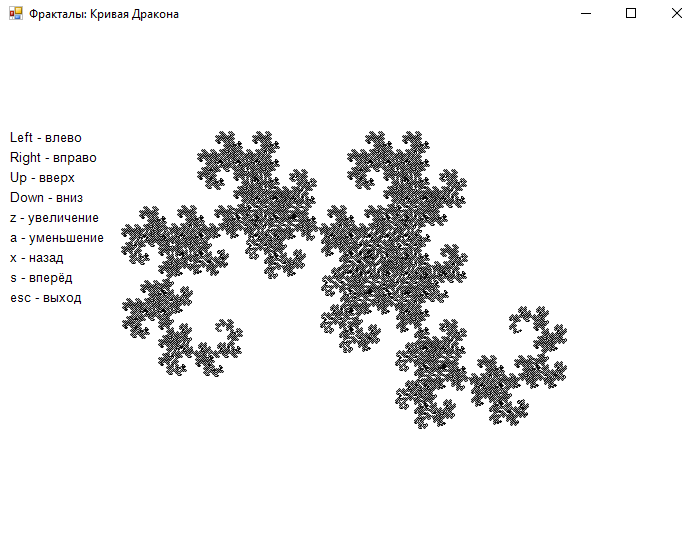
****

Рис. 4 «Результат выполнения программы»

**Вывод**

В процессе выполнения данной работы мы успешно применили изученный материал лабораторных занятий на практике, включая процедуры, условные операторы, циклы создание переменных и операторов ввода-вывода.

Благодаря разработанным программам мы усовершенствовали навыки реализации алгоритмов с применением рекурсивных вычислений. Мы изучили различные виды фракталов и научились создавать их с помощью рекурсии. Также мы освежили знания по работе с GraphABC и научились разделять блоки программы на отдельные модули для упрощения чтения и оптимизации кода. Мы научились использовать виртуальные клавиши для управления фигурами и добавили функционал изменения масштаба и глубины прорисовки изображения. Эти навыки будут полезны нам при работе с GraphABC и управлении клавишами в будущем.