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 ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Калинин Максим Константинович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**Вариант 11**

**Задача:**

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

**Описание фрактала:**

**Кривая Хартера-Хейтуэя** (кривая дракона) – фрактальная кривая, которая определяется рекурсивно и имеет вид траектории, начерченной вдоль линий квадратной решетки.

Кривая дракона имеет следующие характеристики:

* Самопересечения: кривая не имеет точек самопересечения, что делает ее уникальной среди фрактальных кривых
* Порядок: кривая может быть определена для различных порядков, что определяет количество звеньев получающейся ломаной
* Алгоритмы: существуют разные алгоритмы для построения кривой дракона, такие как использование **последовательности** двоичных цифр, складываемой полоски бумаги и геометрического построения
* Предельная фрактальная кривая: при стремлении порядка к бесконечности, кривая Хартера-Хейтуэя превращается в предельную фрактальную кривую

Своё название получило из-за сходства с драконом с когтистыми лапами и развернутой пастью, обращенной влево

Алгоритм построения фрактала кривой Хартера-Хейтуэя:

1. Разделить произвольно заданную прямую пополам
2. Согнуть под углом 90°
3. Выполнение пунктов 1–2 до тех пор, пока указанный счетчик не станет равен нулю
4. Как только счетчик станет равен нулю, рекурсивно построить драконью кривую

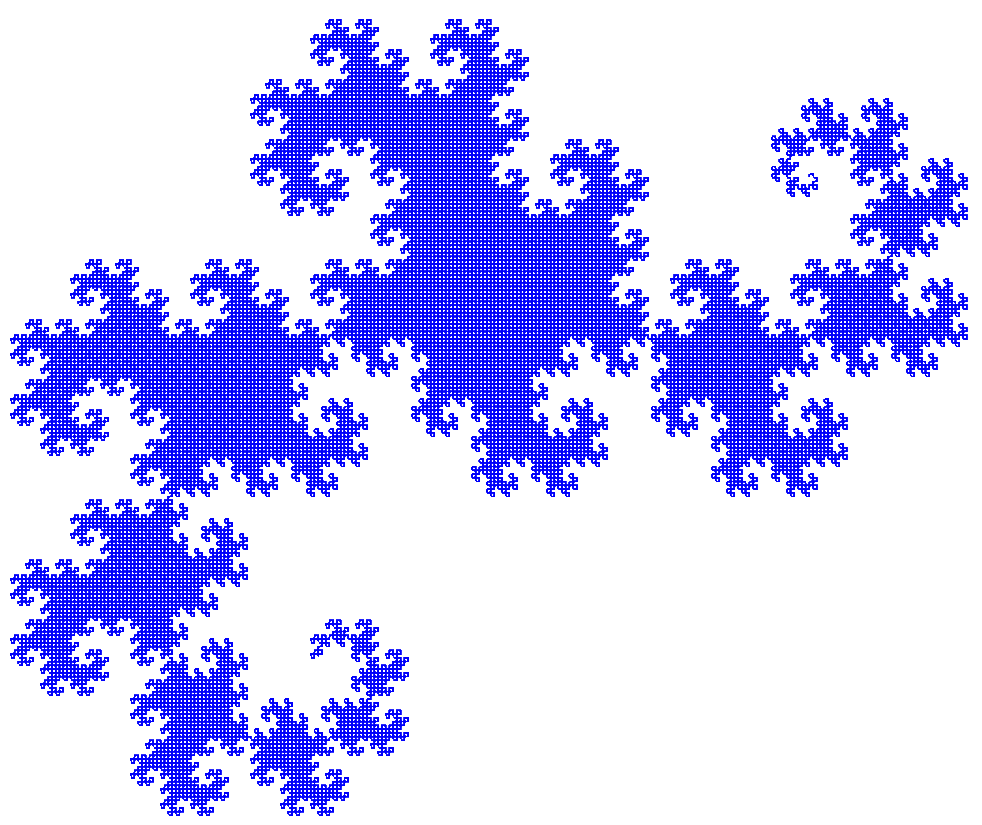


Рис. 0 – Кривая Хартера-Хейтуэя

**Алгоритм решения**



Рис.1 - Алгоритм задания

**Описание алгоритма**

1. Открывается графическое окно для визуализации фрактала "Кривая Хартера-Хейтузя";
2. Инициализируются переменные для масштабирования (scale), глубины прорисовки (depth), начального положения по оси x и по оси y;
3. Устанавливается размер и заголовок окна;
4. Задаются обработчики нажатия клавиш для управления перемещением (w, a, s, d), масштабированием (r, f) и изменением глубины прорисовки (q, e);
5. Инициализируется модуль для построения фрактала;
6. Определяется процедура kurwa для рекурсивного построения фрактала.
7. Если глубина прорисовки равна 0, рисуется линия от точки (x1, y1) до точки (x2, y2);
8. В противном случае, определяются координаты середины отрезка и вызывается процедура kurwa для двух новых отрезков, полученных из исходного путем поворота на 90 градусов и сжатия;
9. При нажатии клавиш происходит изменение соответствующих переменных (перемещение, масштабирование, изменение глубины прорисовки) и перерисовка фрактала с новыми параметрами.

**Код решения программы**

**Главная программа:**

**uses** GraphABC, sus;

//q,e - глубина

//w,a,s,d - сдвиг

//r,f - vfcinf,

**var**

scale:real=1;

depth:integer=4;

x:real=0;

y:real=0;

start\_x:integer=200;

start\_y:integer=400;

end\_x:integer=700;

end\_y:integer=400;

**procedure** redraw;

**begin**

ClearWindow;

kurwa(round(start\_x\*scale+x), round(start\_y\*scale+y), round(end\_x\*scale+x), round(end\_y\*scale+y), depth);

**end**;

**procedure** kay(key:integer);

**begin**

**case** key **of**

VK\_s: **begin** y:=y-50; redraw; **end**;

VK\_w: **begin** y:=y+50; redraw; **end**;

VK\_d: **begin** x:=x-50; redraw; **end**;

VK\_a: **begin** x:=x+50; redraw; **end**;

VK\_r: **begin** scale:=scale\*2; redraw; **end**;

VK\_f: **begin** scale:=scale/2; redraw; **end**;

VK\_e: **begin** depth:=depth+1; redraw; **end**;

VK\_q: **if** depth>0 **then begin** depth:=depth-1; redraw; **end**;

**end**;

**end**;

**begin**

SetWindowSize(800, 800);

SetWindowCaption('Фрактал: Кривая дракона');

OnKeyDown:=kay;

redraw;

**end**.

**Модуль:**

**unit** sus;

**interface**

**uses** graphABC;

**procedure** kurwa(x1,y1,x2,y2,depth: integer);

**implementation**

**procedure** kurwa(x1,y1,x2,y2,depth: integer);

**begin**

**if** depth=0 **then**

line(x1,y1,x2,y2)

**else**

**begin**

**var** midX:=(x1+x2) **div** 2+(y2-y1) **div** 2;

**var** midY:=(y1+y2) **div** 2-(x2-x1) **div** 2;

kurwa(x1,y1,midX,midY, depth-1);

kurwa(x2,y2,midX,midY, depth-1);

**end**;

**end**;

**end**.

**Результат выполнения программы**

**Изображение выглядит как снимок экрана, диаграмма, Прямоугольник, линия

Автоматически созданное описание**

Рис.2 – Первоначальный вывод программы

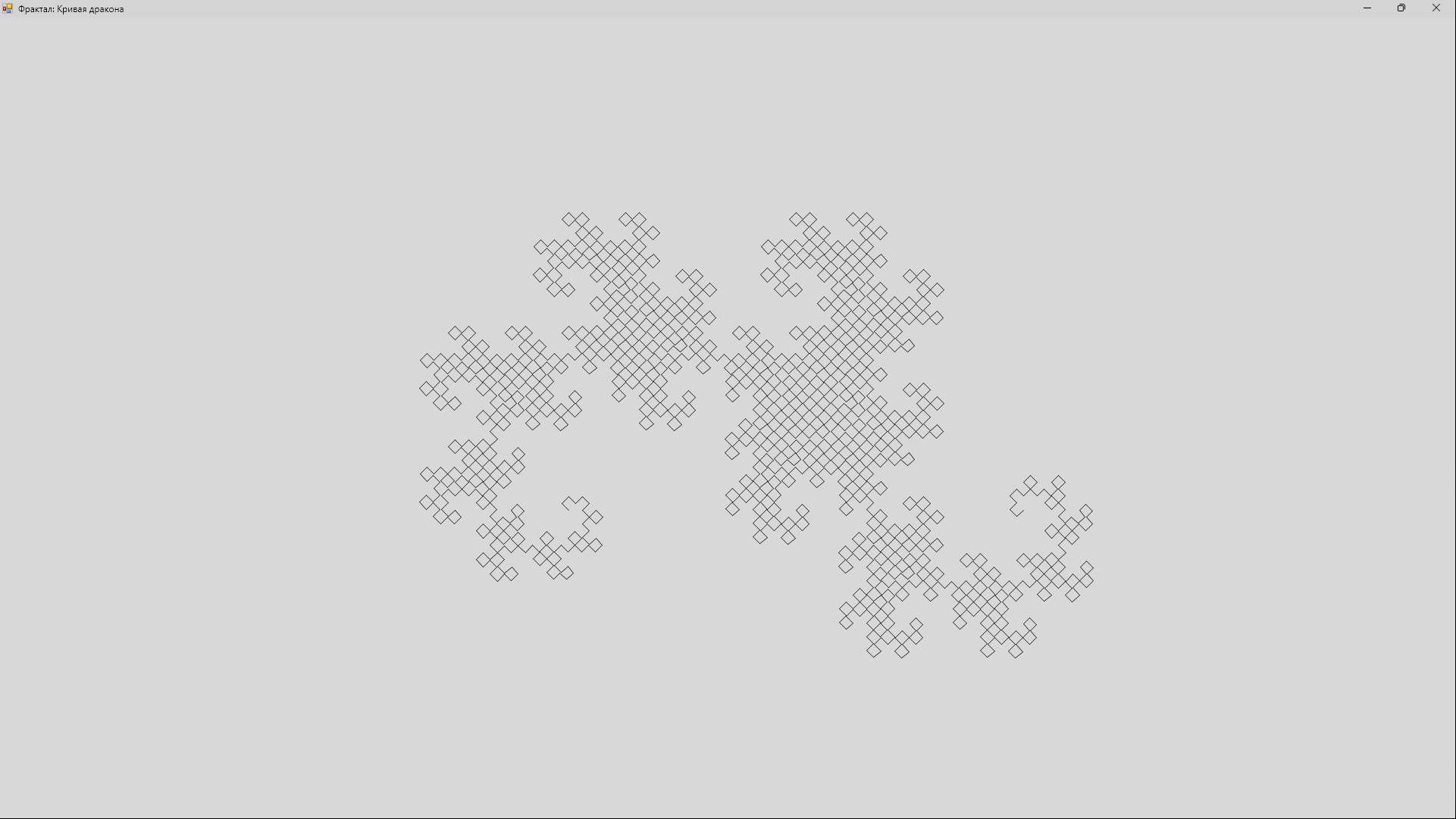
****

Рис.3 – Изменение глубины

Изображение выглядит как рисунок, зарисовка, снежинка, шаблон

Автоматически созданное описание

Рис.4 – Высокое значение глубины

**** Рис.5 – Изменение масштаба

**Вывод:**

1. Овладел алгоритмом построения фрактала "Кривая Хартера-Хейтузя".
2. Разработал модуль для построения фрактала, что научило использовать его для разделения программы.
3. Улучшил понимание принципов рекурсивных алгоритмов, необходимых для построения фракталов.
4. Овладел техниками управления перемещением, масштабированием и изменением глубины прорисовки фрактала, функциональности в программу.
5. Расширил свои знания о возможностях языка Pascal, применяя его для создания графических приложений.
6. Объединил графические элементы и алгоритмы обработки данных.
7. Понял важность структурирования кода и создания модульных компонентов для облегчения его поддержки и дальнейшего развития.
8. Улучшил навыки отладки программы, особенно при работе с графическим интерфейсом, что помогло сделать программу более стабильной и функциональной.