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

Кривошеин Дмитрий Васильевич

Преподаватель:

Сергеева Елизавета Григорьевна

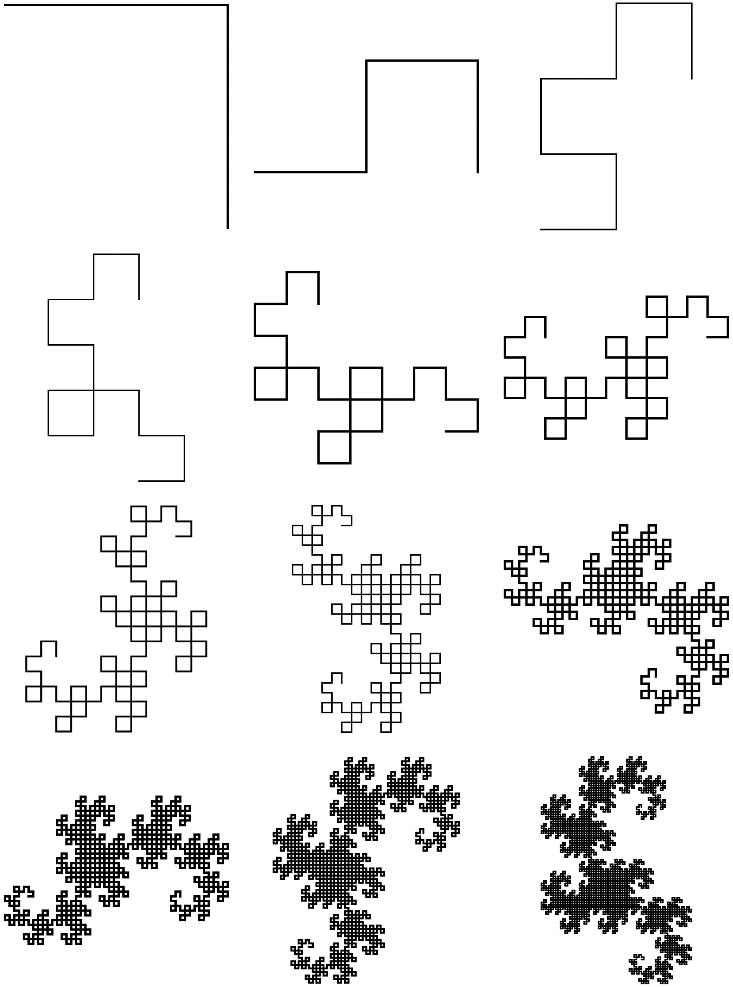
г. Киров

2024

1. **Цель работы**: получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.
2. **Задание:**
3. Написать программу для визуализации фрактала “Кривая Хартера­Хейтуэя”.
4. Предусмотреть возможность масштабирования, изменения глубины прорисовки и перемещение полученной фигуры.
5. Построение множества ломанных, образующих фрактал, должно осуществляться в отдельном модуле.
6. **Описание алгоритма:**

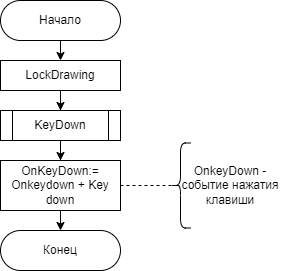
Фрактал “Кривая Хартера-Хейтуэя” представляет собой геометрический фрактал, который впервые описан Гарри Хартером и Полом Хейтуэем в 1967 году. Этот фрактал обладает свойством самоподобия – то есть, его структура повторяется на разных масштабах. Кривая Хартера-Хейтуэя часто используется в компьютерной графике в визуализации данных, а также в научных исследованиях.

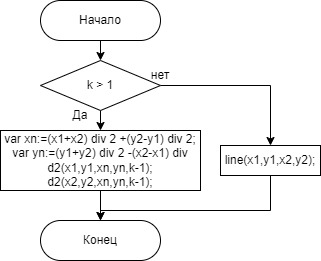
Берём отрезок, сгибаем его пополам. Затем многократно повторяем итерацию. Если после этого снова разогнуть получившуюся (сложенную) линию так, чтобы все углы были равны 90°, мы получим драконову ломаную.



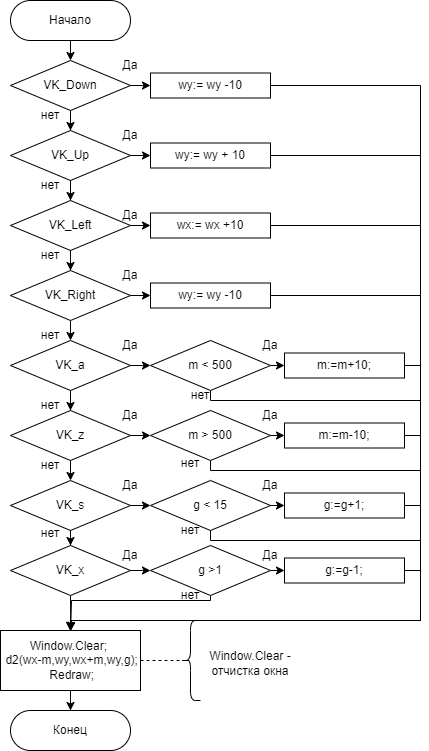
**Рисунок 1 – схема преобразования фрактала**

1. **Схема алгоритма:**

 **Рисунок 2 – алгоритм решения**



**Рисунок 3 -модуль risov**



**Рисунок 4 – подпрограмма KeyDown**

**Код программ:**

**Uses** GraphABC;

**Uses** risov;

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

**begin**

**case** K **of**

VK\_Down: wy := wy-10;

VK\_Up: wy := wy+10;

VK\_Left: wx := wx+10;

VK\_Right: wx := wx-10;

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

VK\_z: **if** m > 100 **then** m:=m-10;

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

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

**end**;

Window.Clear;

d2(wx-m,wy,wx+m,wy,g);

Redraw;

**end**;

**Begin**

LockDrawing;

KeyDown(0);

OnkeyDown += KeyDown;

**End**.

**Модуль risov:**

**Unit** risov;

**Uses** GraphABC;

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

**Procedure** d2(x1,y1,x2,y2,k:integer); – параметризация

**begin**

**If** k>1 **then begin**

**var** xn:=(x1+x2) **div** 2 +(y2-y1) **div** 2; – декомпозиция

**var** yn:=(y1+y2) **div** 2 -(x2-x1) **div** 2; – декомпозиция

d2(x1,y1,xn,yn,k-1);

d2(x2,y2,xn,yn,k-1);

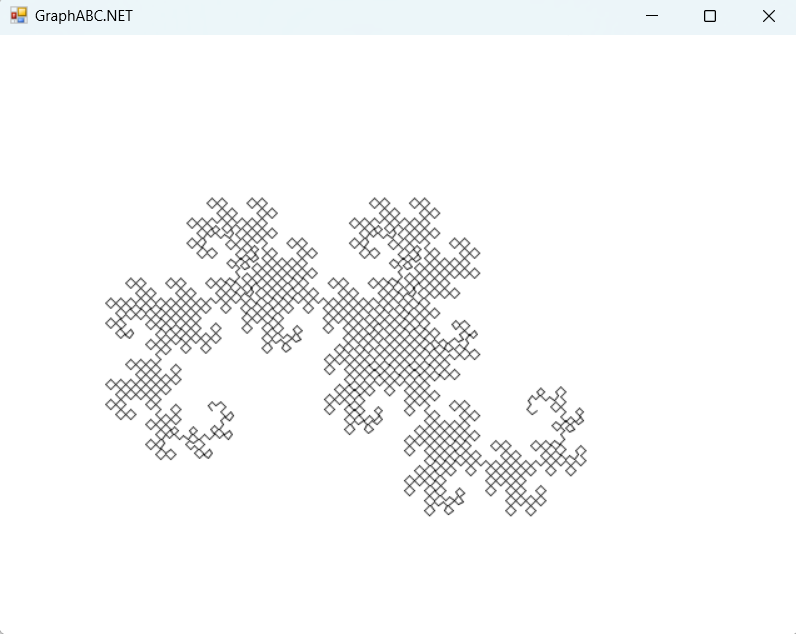
**end**

**else** line(x1,y1,x2,y2); – база рекурсии

**end**;

**end**.

**Результат выполнения программ:**



1. **Вывод:**

Программы были успешно разработаны и выполнены согласно поставленной задаче. Вот основные выводы:

Мы получили навыки реализации алгоритмов рекурсивными вычислениями. Познакомились с фракталами разных видов и их созданием с помощью рекурсии. Освежили свои знания в работе с GraphABC, а так же научились выводить блок программы в отдельный модуль, что в будущем может помочь нам для оптимизации кода, а так е для упрощения его чтения. Научились использовать виртуальные коды клавиш для перемещения фигуры по рабочему пространству, а так же это помогла нам произвести возможность масштабирования и изменения глубины прорисовки изображения. Это может помочь нам в дальнейшем при работе с модулем GraphABC и работе с клавишами.