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**

**«ИССЛЕДОВАНИ ФРАКТАЛОВ»**

**ПО МДК 05.02 «РАЗРАБОТКА КОДА ИНФЛОРМАЦИОННЫХ СИСТЕМ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Носкова Анастасия Владимировна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. **Цель работы**

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

1. **Формулировка задания**

Вариант: 16

1. Написать программу для визуализации фрактала «Кривая Гильберта».
2. Предусмотреть возможности масштабирования, изменения глубины прорисовки и перемещения полученной фигуры.
3. Построение множества ломаных, образующих фрактал, должно осуществляться в отдельном модуле.
4. **Описание фрактала**

Кривая Гильберта – это непрерывная фрактальная кривая, образуется путем соединения последовательности U-образных кривых, расположенных и ориентированных в разных направлениях. При увеличении масштаба можно увидеть, что она выглядит так же, как сама кривая. Была впервые описана Давидом Гильбертом в 1891 году.

Ломаная-генератор для кривой Гильберта – это U-образная кривая.

Алгоритм построения:

* Начало – линия-генератор.
* На каждом шаге заменяем каждый сегмент ломанной на меньшую копию всей ломанной, повернутую и отраженную таким образом, чтобы она соединяла концы сегмента.
* Повторяем предыдущее действие столько раз, сколько указано в глубине фрактала.

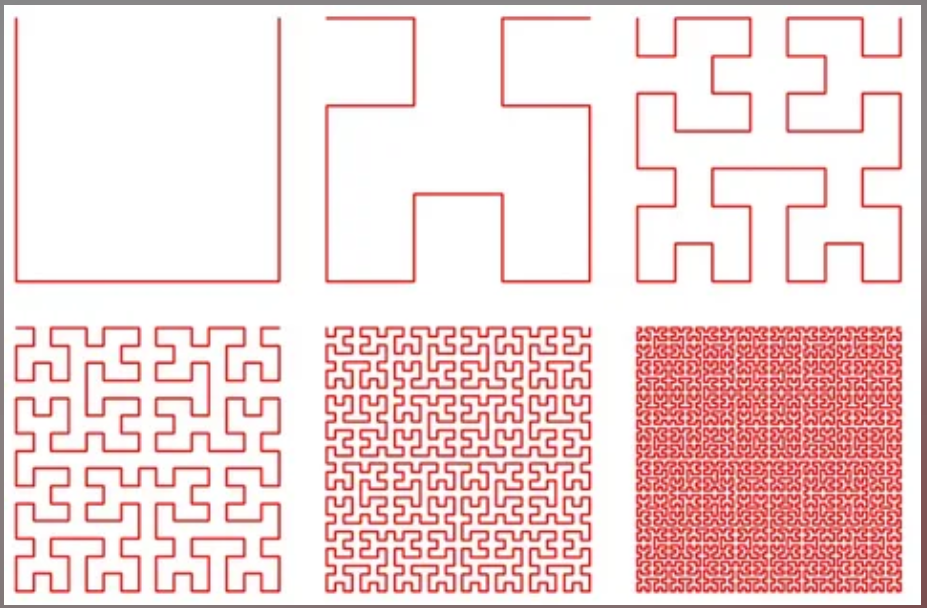


Рисунок 1 – Кривая гильберта с разным уровнем глубины

1. **Схема алгоритма**

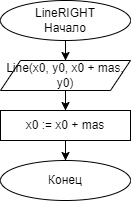
****

Рисунок 1 – Схема процедуры построения линий вправо

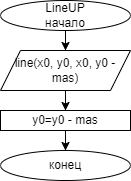
****

Рисунок 2 – Схема процедуры построения линий вверх

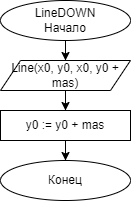
****

Рисунок 3 – Схема процедуры построения линий вниз

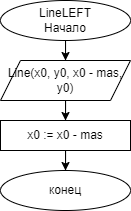
****

Рисунок 4 – Схема процедуры построения линий влево

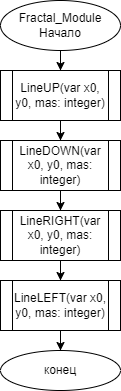


Рисунок 5 – Схема программы модуля

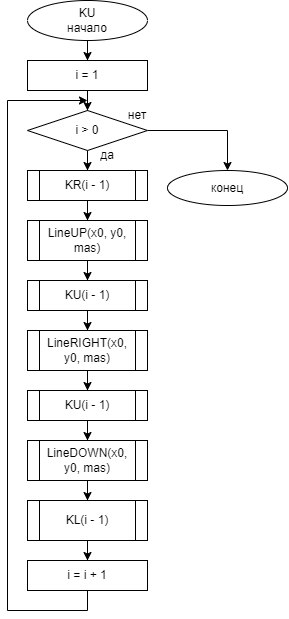


Рисунок 6 – Схема функции для рисования линий вверх

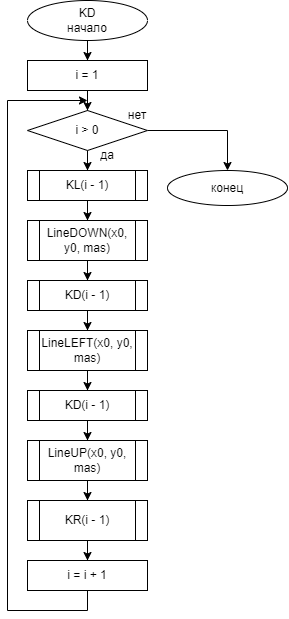


Рисунок 7 – Схема алгоритма функции рисований линий вниз

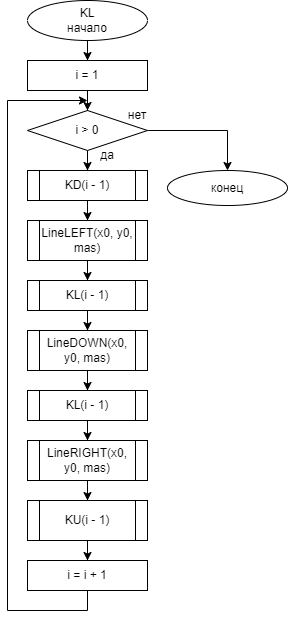


Рисунок 8 – Схема алгоритма функции рисований линий влево

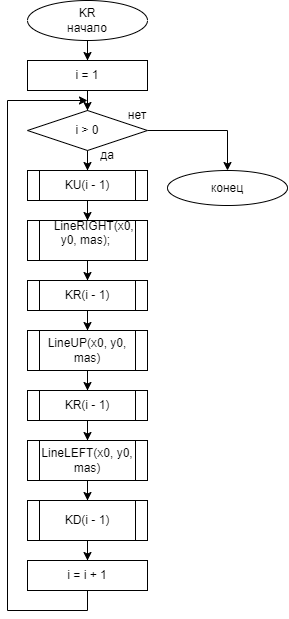


Рисунок 9 – Схема алгоритма функции рисования линий вправо

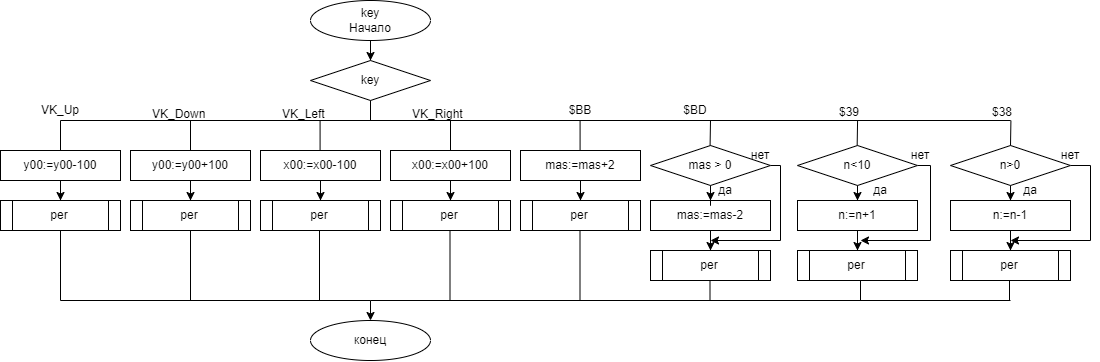
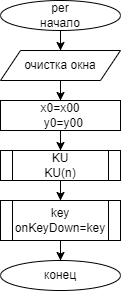


Рисунок 10 – Схема алгоритма процедуры обработки нажатий клавиш



Рисование 11 – Схема алгоритма функции начала рисования фрактала

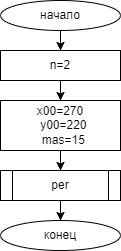


Рисунок 12 – Схема алгоритма главной программы

1. **Код программы**
2. **Программа**

**uses** GraphABC, Fractal\_Module;

**var**

x0, y0, x00, y00, mas, n: integer;

**function** KU(i:integer):integer; **forward**;

**function** KD(i:integer):integer; **forward**;

**function** KR(i:integer):integer; **forward**;

**function** KL(i:integer):integer; **forward**;

**function** per: integer; **forward**;

// Функция для рисования линии вверх

**function** KU( i: integer): integer;

**begin**

**if** i > 0 **then**

**begin**

KR(i - 1);

LineUP(x0, y0, mas);

KU(i - 1);

LineRIGHT(x0, y0, mas);

KU(i - 1);

LineDOWN(x0, y0, mas);

KL(i - 1);

**end**;

**end**;

// Функция для рисования линии вниз

**function** KD( i: integer): integer;

**begin**

**if** i > 0 **then**

**begin**

KL(i - 1);

LineDOWN(x0, y0, mas);

KD(i - 1);

LineLEFT(x0, y0, mas);

KD(i - 1);

LineUP(x0, y0, mas);

KR(i - 1);

**end**;

**end**;

// Функция для рисования линии влево

**function** KL( i: integer): integer;

**begin**

**if** i > 0 **then**

**begin**

KD(i - 1);

LineLEFT(x0, y0, mas);

KL(i - 1);

LineDOWN(x0, y0, mas);

KL(i - 1);

LineRIGHT(x0, y0, mas);

KU(i - 1);

**end**;

**end**;

// Функция для рисования линии вправо

**function** KR( i: integer): integer;

**begin**

**if** i > 0 **then**

**begin**

KU(i - 1); // Рекурсивный вызов функции для рисования линии вверх

LineRIGHT(x0, y0, mas); // Рисование линии вправо

KR(i - 1); // Рекурсивный вызов функции для рисования линии вправо

LineUP(x0, y0, mas); // Рисование линии вверх

KR(i - 1); // Рекурсивный вызов функции для рисования линии вправо

LineLEFT(x0, y0, mas); // Рисование линии влево

KD(i - 1); // Рекурсивный вызов функции для рисования линии вниз

**end**;

**end**;

// Процедура для обработки нажатий клавиш

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

**begin**

**case** key **of**

VK\_Up : **begin** y00:=y00-100; per; **end**;

VK\_Down : **begin** y00:=y00+100; per; **end**;

VK\_Left : **begin** x00:=x00-100; per; **end**;

VK\_Right : **begin** x00:=x00+100; per; **end**;

$BB : **begin** mas:=mas+2; per; **end**; // Увеличение размера фрактала

$BD : **begin if** mas > 0 **then** mas:=mas-2; per; **end**; // Уменьшение размера фрактала

$39 : **begin if** n<10 **then** n:=n+1; per; **end**; // Увеличение уровня рекурсии

$38 : **begin if** n>0 **then** n:=n-1; per; **end**; // Уменьшение уровня рекурсии

**end**;

**end**;

// Функция для начала рисования фрактала

**function** per: integer;

**begin**

**var** center := window.center;

clearwindow;

x0:=x00; // Установка начальных координат

y0:=y00;

KU(n); // Начало рисования фрактала

onKeyDown:=key; // клавиши

redraw;

**end**;

**begin**

n:=2; // начальные значения переменных

x00:=270;

y00:=220;

mas:=15;

lockdrawing;

per // Вызов функции для начала рисования фрактала

**end**.

1. **Модуль программы**

**unit** Fractal\_Module;

**interface**

**uses** GraphABC;

**procedure** LineUP(**var** x0, y0, mas: integer);

**procedure** LineDOWN(**var** x0, y0, mas: integer);

**procedure** LineRIGHT(**var** x0, y0, mas: integer);

**procedure** LineLEFT(**var** x0, y0, mas: integer);

**implementation**

**procedure** LineUP(**var** x0, y0, mas: integer);

**begin**

line(x0, y0, x0, y0 - mas);

y0 := y0 - mas;

**end**;

**procedure** LineDOWN(**var** x0, y0, mas: integer);

**begin**

Line(x0, y0, x0, y0 + mas);

y0 := y0 + mas;

**end**;

**procedure** LineRIGHT(**var** x0, y0, mas: integer);

**begin**

Line(x0, y0, x0 + mas, y0);

x0 := x0 + mas;

**end**;

**procedure** LineLEFT(**var** x0, y0, mas: integer);

**begin**

Line(x0, y0, x0 - mas, y0);

x0 := x0 - mas;

**end**;

**end**.

1. **Результат выполнения программы:**

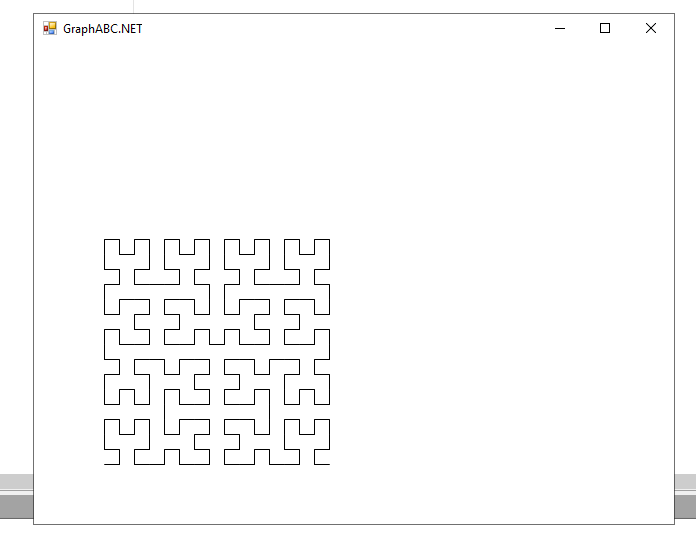


Рисунок 2 – Результаты выполнение программы 1

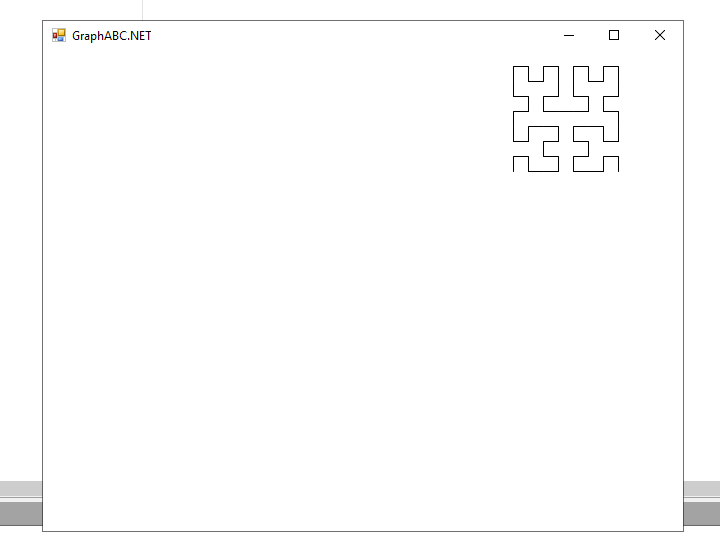


Рисунок 3 – Результаты выполнение программы 2

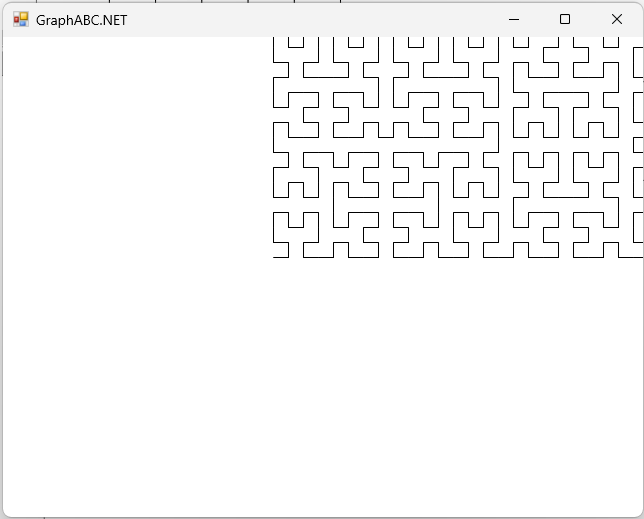


Рисунок 4 – Результаты выполнения программы 3

Изображение выглядит как Прямоугольник, снимок экрана, диаграмма, прямоугольный

Автоматически созданное описание

Рисунок 5 -Результаты выполнения программы 4

1. **Вывод**

В результате выполнения работы мы разобрались что такое фракталы. Подробно разобрали фрактал «Кривая Гильберта», что он из себя представляет, как выглядит и как строится.

Была написана программа для визуализации фрактала «Кривая Гильберта», также предусмотрены возможности масштабирования, изменения глубины прорисовки и перемещения полученной фигуры. Построение множества ломанных, образующих фрактал осуществляется в отдельном модуле.