Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №1**

**«Исследование фракталов»**

**ПО «МДК.05.02 Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Макаров Никита Андреевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

**Цель работы:**

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

**Формулировка задания:**

* Написать программу для визуализации фрактала «Кривая Хартера-Хейтуэя».
* Предусмотреть возможность масштабирования, изменения глубины прорисовки и перемещения полученной фигуры.
* Построение множества ломанных, образующих фрактал, должно осуществляться в отдельном модуле.

**Описание алгоритма и ответы на вопросы:**

Алгоритм:

1. Инициализация окна:

-Устанавливается размер окна (`SetWindowSize(800, 600)`).

-Задается заголовок окна: SetWindowCaption('Кривая Хартера-Хетуэя')

2. Основная процедура DrawHarterHeighway

-Она рекурсивно рисует кривую Хартера

-Параметры:

- x,y текущие координаты

-scale масштаб

-depth глубина рекурсии( кол-во итераций)

3.Рекурсивное построение кривой Хартера

-если depth = 0 тогда,

-строиться отрезок(x1,y1,x2,y2)

-Повернуть его на 90 градусов вокруг одной из вершин и добавить полученный отрезок к исходному. Получится уголок из двух отрезков.

-Повторяется процедура

-Повернуть уголок на 90 градусов вокруг вершины и добавить полученную ломаную к исходной.

4. Вспомогательная процедура Dragon

 Сначала задается переменная xn по формуле Round((x1 + x2) / 2 + (y2 - y1) / 2 \* angle)

После этого задается yn по формуле Round((y1 + y2) / 2 - (x2 - x1) / 2 \* angle)

Строиться 2 отрезка по x и y Dragon(x1, y1, xn, yn, depth - 1);

Dragon(x2, y2, xn, yn, depth - 1)

После этого находится sin p/4 и строиться окончательный отрезок

Dragon(Round(x - 100 \* scale), Round(y - 50 \* scale), Round(x + 100 \* scale), Round(y - 50 \* scale), depth)

5. Ожидание действий пользователя с интерфейсом: масштабирования /движение/изменение глубины

**Схемы алгоритмов**:

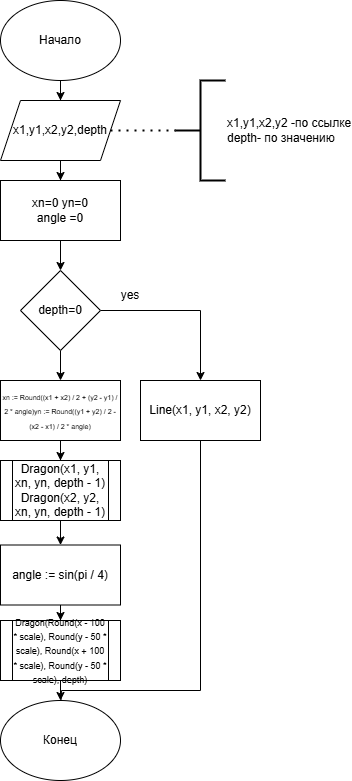


Рисунок 1- схема алгоритма процедуры Dragon в модуле FractalModule

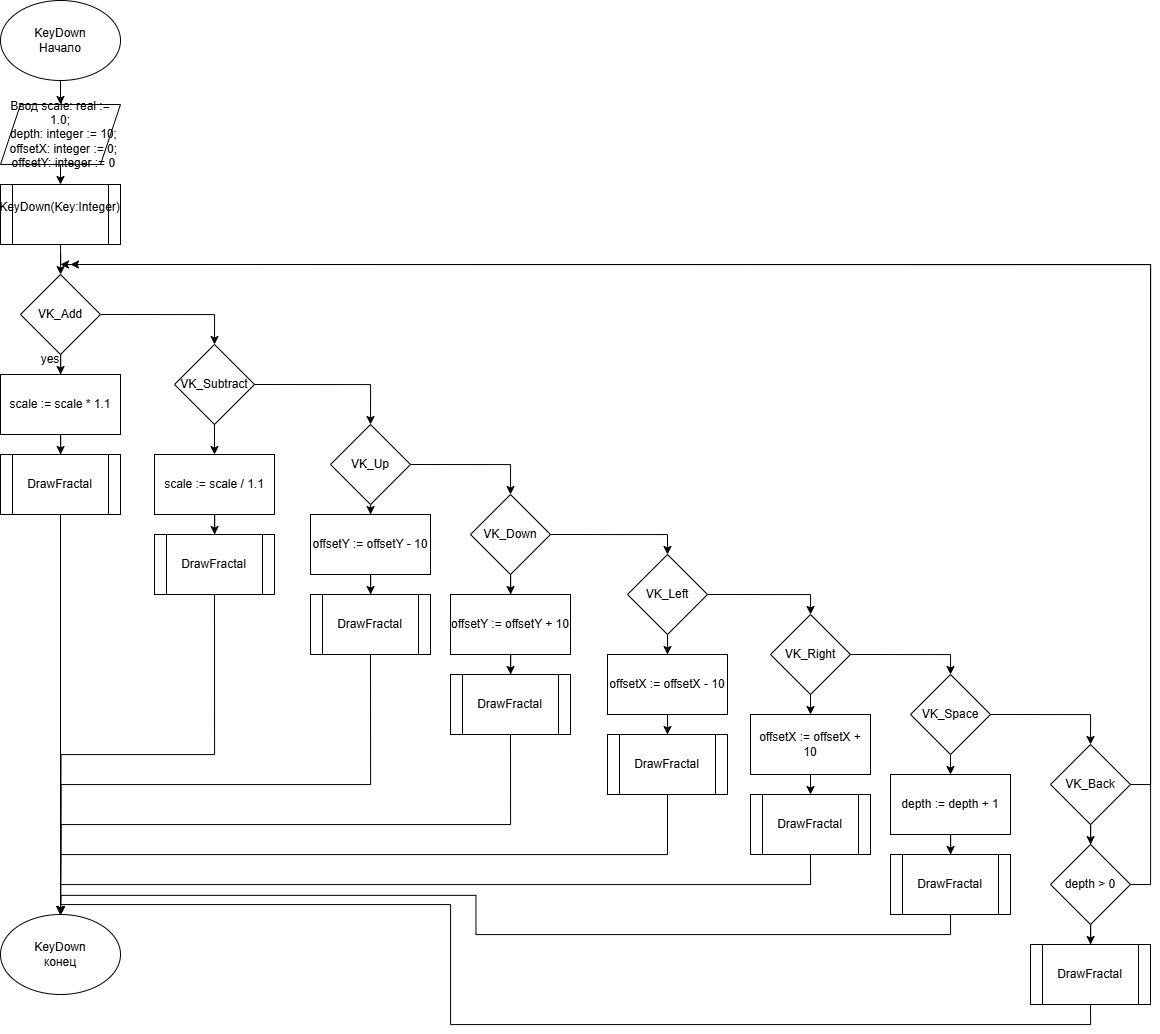


Рисунок 2 – схема алгоритма процедуры KeyDown

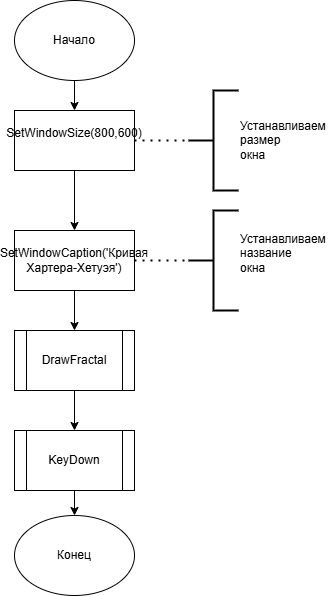


Рисунок 3 – схема алгоритма основной программы

**Код программы:**

**program HarterHeighwayDragon;**

**uses**

**GraphABC, FractalModule;**

**var**

**scale: real := 1.0;**

**depth: integer := 10;**

**offsetX: integer := 0;**

**offsetY: integer := 0;**

**procedure DrawFractal;**

**begin**

**ClearWindow;**

**FractalModule.DrawHarterHeighway(WindowWidth div 2 + offsetX, WindowHeight div 2 + offsetY, scale, depth);**

**end;**

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

**begin**

**case Key of**

**VK\_Add: // Увеличение масштаба**

**begin**

**scale := scale \* 1.1;**

**DrawFractal;**

**end;**

**VK\_Subtract: // Уменьшение масштаба**

**begin**

**scale := scale / 1.1;**

**DrawFractal;**

**end;**

**VK\_Up: // Перемещение вверх**

**begin**

**offsetY := offsetY - 10;**

**DrawFractal;**

**end;**

**VK\_Down: // Перемещение вниз**

**begin**

**offsetY := offsetY + 10;**

**DrawFractal;**

**end;**

**VK\_Left: // Перемещение влево**

**begin**

**offsetX := offsetX - 10;**

**DrawFractal;**

**end;**

**VK\_Right: // Перемещение вправо**

**begin**

**offsetX := offsetX + 10;**

**DrawFractal;**

**end;**

**VK\_Space: // Увеличение глубины прорисовки**

**begin**

**depth := depth + 1;**

**DrawFractal;**

**end;**

**VK\_Back: // Уменьшение глубины прорисовки**

**begin**

**if depth > 0 then**

**depth := depth - 1;**

**DrawFractal;**

**end;**

**end;**

**end;**

**begin**

**SetWindowSize(800, 600);**

**SetWindowCaption('Кривая Хартера-Хетуэя');**

**OnKeyDown := KeyDown;**

**DrawFractal;**

**end.**

**unit FractalModule;**

**interface**

**uses**

**GraphABC;**

**procedure DrawHarterHeighway(x, y: integer; scale: real; depth: integer);**

**implementation**

**procedure DrawHarterHeighway(x, y: integer; scale: real; depth: integer);**

**var**

**angle: real;**

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

**var**

**xn, yn: integer;**

**begin**

**if depth = 0 then**

**begin**

**Line(x1, y1, x2, y2);**

**end**

**else**

**begin**

**xn := Round((x1 + x2) / 2 + (y2 - y1) / 2 \* angle);**

**yn := Round((y1 + y2) / 2 - (x2 - x1) / 2 \* angle);**

**Dragon(x1, y1, xn, yn, depth - 1);**

**Dragon(x2, y2, xn, yn, depth - 1);**

**end;**

**end;**

**begin**

**angle := sin(pi / 4);**

**Dragon(Round(x - 100 \* scale), Round(y - 50 \* scale), Round(x + 100 \* scale), Round(y - 50 \* scale), depth);**

**end;**

**end.**

**Результат выполнения программы:**

**Изображение выглядит как снимок экрана, текст

Контент, сгенерированный ИИ, может содержать ошибки.**

**Рисунок – результат выполнения программы**

**Изображение выглядит как снимок экрана, текст

Контент, сгенерированный ИИ, может содержать ошибки.**

**Рисунок – результат выполнения одной из кнопок**

**Вывод:**

В ходе работы был изучен алгоритм построения фрактала «Кривая Хартера-Хейтуэя», её реализация на программном коде, также были реализованы такие функции как масштабирование, изменение глубины, а также перемещение полученной фигуры вдоль осей x и y. Был получен новый опыт с модулем GraphABC ,а также с рекурсивными функциями при построении фракталов.