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 «МДК 05.02 Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-206-52-00

Долинин Владислав Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

В отчете должны отображаться:

1. Цель работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма и ответы на вопросы
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод
8. Цель работы - получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.
9. Вариант 12.

1. Написать программу для визуализации фрактала "Кривая Хартера-Хейтуэя".

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

3. Построение множества ломанных, образующих фрактал, должно осуществляться в отдельном модуле.

1. Описание алгоритма и ответы на вопросы

Программа 1.

depth — глубина рекурсии.

scale — масштаб изображения.

offsetX и offsetY — смещение кривой по осям X и Y.

VK\_Add, VK\_PageUp — увеличивают глубину.

VK\_Subtract, VK\_PageDown — уменьшают глубину.

VK\_Up, VK\_Down, VK\_Left, VK\_Right — сдвигают кривую по осям X и Y.

Функция DrawDragonCurve генерирует и рисует кривую в зависимости от параметров.

Размер окна: 800x600 пикселей.

Заголовок: 'Кривая Хартера-Хейтуэя'.

Программа 2.

DragonCurve — рекурсивная функция, генерирующая саму кривую дракон.

DrawDragonCurve — основная процедура для рисования.

depth — глубина рекурсии, определяющая сложность кривой.

scale — масштаб, определяющий размер кривой.

offsetX, offsetY — смещение, сдвигающее кривую по осям X и Y.

Очищение окна.

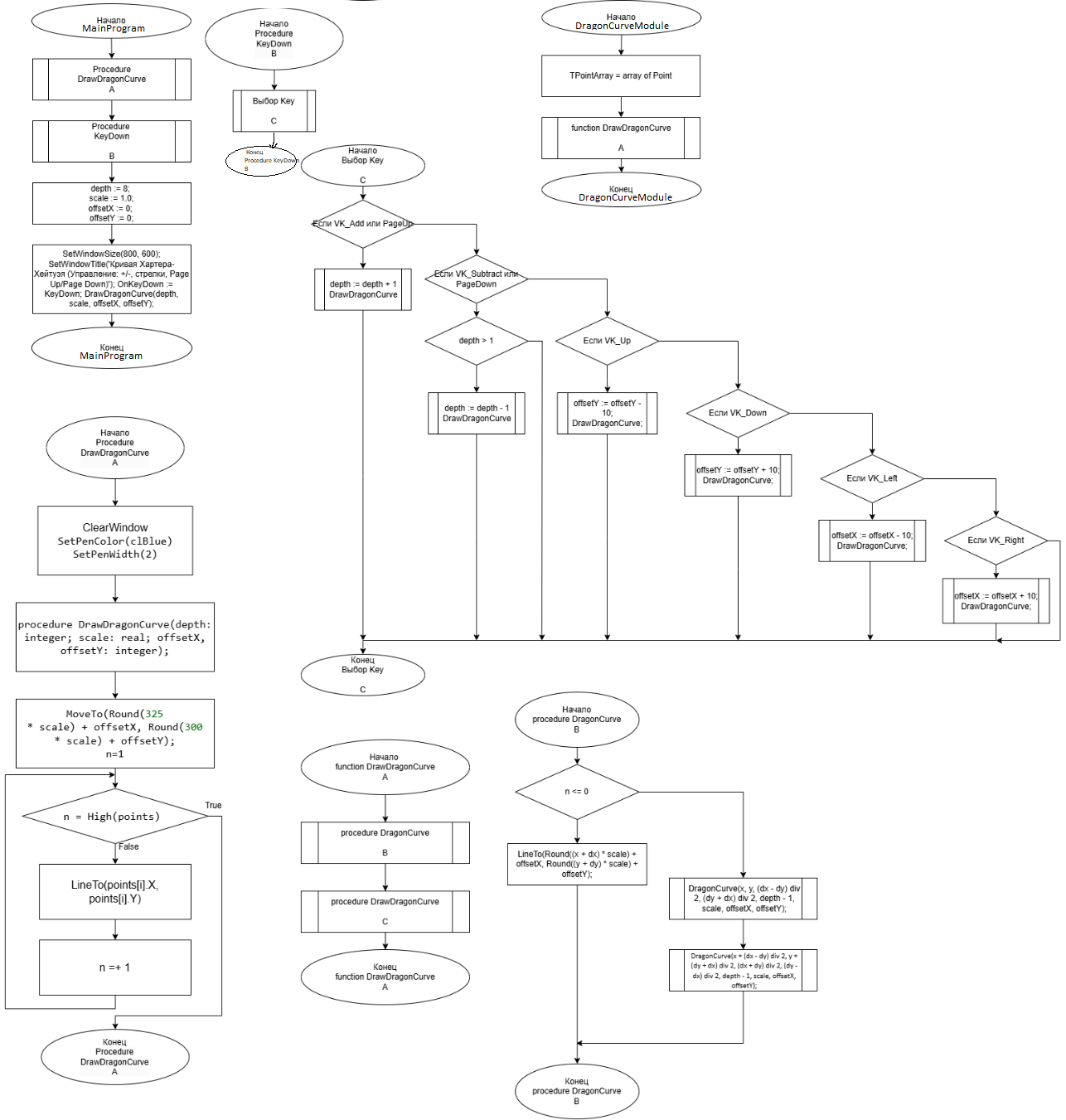
Перемещение в начальную точку.

Рисование кривой дракона с использованием рекурсии.

Ответы на вопросы:

1. Фрактал – это фигура, обладающая свойством самоподобия   
2. Кривая Хартера-Хейтуэя строится из отрезка, который на каждом шаге заменяется на два новых, повернутых под углом 90 градусов. С каждым шагом кривая становится всё более извилистой и сложной, но никогда не пересекает саму себя. Этот фрактал демонстрирует, как простые правила могут создавать удивительно красивые и сложные формы.

1. Схема алгоритма с комментариями



1. Код программы

Программа 1.   
**uses**

GraphABC, DragonCurveModule;

**var**

depth: integer;

scale: real;

offsetX, offsetY: integer;

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

**begin**

**case** Key **of**

VK\_Add, VK\_PageUp:

**begin**

depth := depth + 1;

DrawDragonCurve(depth, scale, offsetX, offsetY);

**end**;

VK\_Subtract, VK\_PageDown:

**begin**

**if** depth > 0 **then**

depth := depth - 1;

DrawDragonCurve(depth, scale, offsetX, offsetY);

**end**;

VK\_Up:

**begin**

offsetY := offsetY - 10;

DrawDragonCurve(depth, scale, offsetX, offsetY);

**end**;

VK\_Down:

**begin**

offsetY := offsetY + 10;

DrawDragonCurve(depth, scale, offsetX, offsetY);

**end**;

VK\_Left:

**begin**

offsetX := offsetX - 10;

DrawDragonCurve(depth, scale, offsetX, offsetY);

**end**;

VK\_Right:

**begin**

offsetX := offsetX + 10;

DrawDragonCurve(depth, scale, offsetX, offsetY);

**end**;

**end**;

**end**;

**begin**

depth := 8;

scale := 1.0;

offsetX := 0;

offsetY := 0;

SetWindowSize(800, 600);

SetWindowTitle('Кривая Хартера-Хейтуэя (Управление: +/-, стрелки, Page Up/Page Down)');

OnKeyDown := KeyDown;

DrawDragonCurve(depth, scale, offsetX, offsetY);

**end**.

Программа 2 – отдельный модуль.

**unit** DragonCurveModule;

**interface**

**uses** GraphABC;

**procedure** DrawDragonCurve(depth: integer; scale: real; offsetX, offsetY: integer);

**implementation**

**procedure** DragonCurve(x, y, dx, dy, depth: integer; scale: real; offsetX, offsetY: integer);

**begin**

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

**begin**

LineTo(Round((x + dx) \* scale) + offsetX, Round((y + dy) \* scale) + offsetY);

**end**

**else**

**begin**

DragonCurve(x, y, (dx - dy) **div** 2, (dy + dx) **div** 2, depth - 1, scale, offsetX, offsetY);

DragonCurve(x + (dx - dy) **div** 2, y + (dy + dx) **div** 2, (dx + dy) **div** 2, (dy - dx) **div** 2, depth - 1, scale, offsetX, offsetY);

**end**;

**end**;

**procedure** DrawDragonCurve(depth: integer; scale: real; offsetX, offsetY: integer);

**begin**

ClearWindow;

MoveTo(Round(325 \* scale) + offsetX, Round(300 \* scale) + offsetY);

DragonCurve(325, 300, 200, 0, depth, scale, offsetX, offsetY);

Redraw;

**end**;

**end**.

1. Результат выполнения программы

Изображение выглядит как диаграмма, зарисовка, Прямоугольник, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как зарисовка, диаграмма, белый, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как диаграмма, зарисовка, План, Технический чертеж

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как диаграмма, зарисовка, белый, оригами

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как диаграмма, План, зарисовка, Технический чертеж

Контент, сгенерированный ИИ, может содержать ошибки.

1. Вывод

Получил навыки реализации алгоритмов с рекурсивными вычислениями, познакомился с фракталами. При выполнении работы у меня возник вопрос в бинде клавиш, как правильно делать. Я обратился к интернет-источнику для решения своей проблемы. Написал программу.