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»**

**«МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ»**

Выполнил: студент учебной группы

ИСПк-202-52-00

Долгополов Ярослав Максимович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

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

**Формулировка задания. Вариант 8**

**Постановка задачи**

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.

**Описание алгоритма**

* 1. Устанавливается размер окна с помощью SetWindowSize(1280, 720).
  2. Переменные x и y инициализируются значением 608 и 360, что означает, что кривая будет строится примерно в центре окна. Изначальная глубина будет равна 10.
  3. Далее производится бесконечный цикл:
     1. Стирается содержимое окна.
     2. Строится фрактал с помощью рекурсивной процедуры ris.
     3. Считываются нажатия кнопок:
        1. При нажатии стрелок кривая перемещается по окну.
        2. При нажатии W увеличивается глубина фрактала.
        3. При нажатии S уменьшается глубина фрактала.
        4. При нажатии D увеличивается масштаб изображения.
        5. При нажатии A уменьшается масштаб изображения, только если масштаб не является стандартным.

**Схема алгоритма**

****

Рисунок 1. — код программы

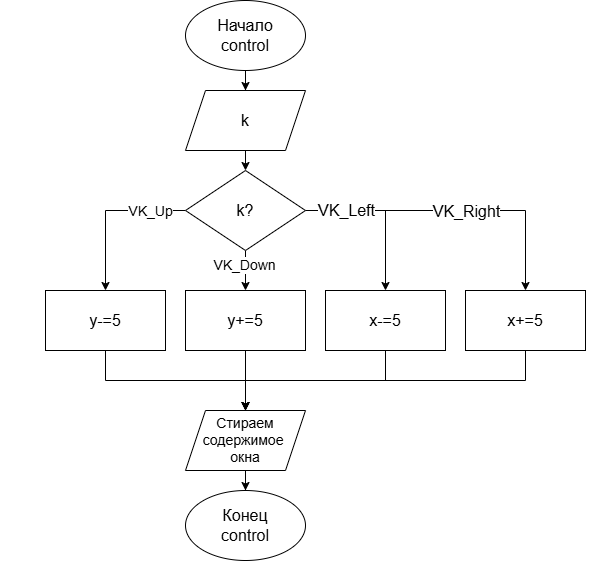


Рисунок 2. — процедура control

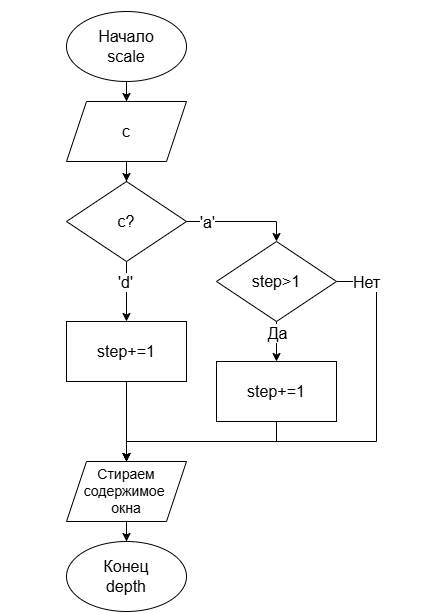


Рисунок 3. — процедура scale

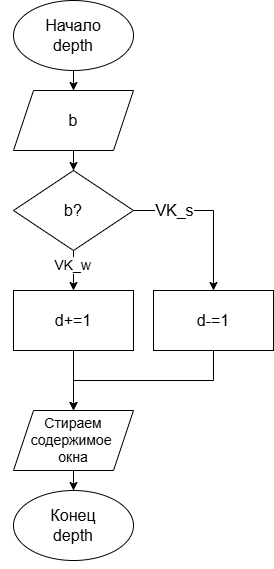


Рисунок 4. — процедура depth

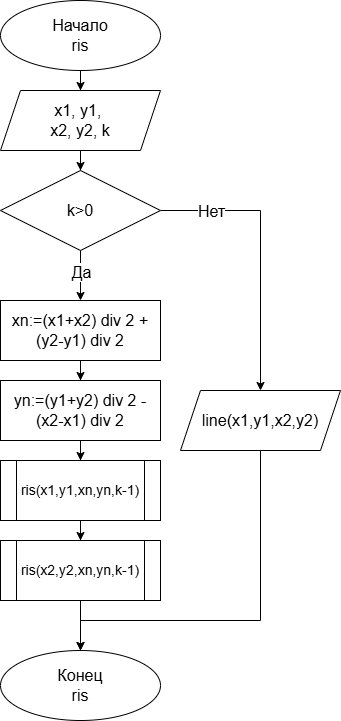


Рисунок 5. — процедура ris

**Код программы**

Основной код

**uses** GraphABC, fractal;

**var**

x, y, d, step, n: integer;

bool: boolean:=false;

**procedure** control(k: integer);

**begin**

**case** k **of**

VK\_Up:

**begin**

y-=5;

**end**;

VK\_Down:

**begin**

y+=5;

**end**;

VK\_Left:

**begin**

x-=5;

**end**;

VK\_Right:

**begin**

x+=5;

**end**;

**end**;

clearwindow

**end**;

**procedure** depth(b:integer);

**begin**

**case** b **of**

VK\_w:d+=1;

VK\_s:d-=1;

**end**;

clearwindow

**end**;

**procedure** scale(c: char);

**begin**

**case** c **of**

'd': step := step + 1;

'a':

**if** step > 1 **then**

step := step - 1;

**end**;

clearwindow

**end**;

**begin**

lockdrawing;

setwindowsize(1280, 720);

d:=10;

step:=1;

x:=608;

y:=360;

**repeat**

**begin**

clearwindow;

ris(x, y, x+64\*step, y, d);

onKeyDown:=control;

onKeyUp:=depth;

onKeyPress:=scale;

redraw;

sleep(1)

**end**;

**until** false

**end**.

Модуль fractal

**unit** fractal;

**uses** GraphABC;

**procedure** ris(x1,y1,x2,y2,k:integer);

**var** xn,yn:integer;

**begin**

**if** k>0 **then**

**begin**

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

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

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

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

**end**

**else begin** line(x1,y1,x2,y2); **end**;

**end**;

**end**.

**Результат выполнения программы**

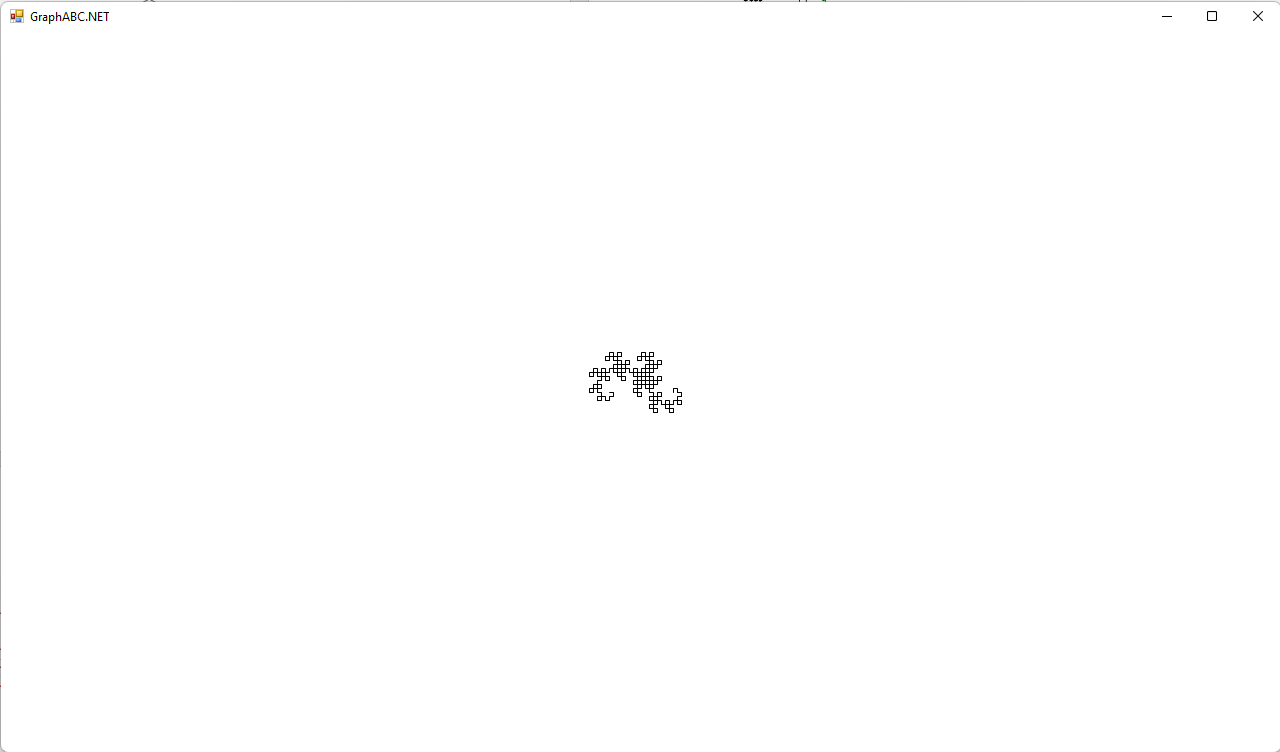
****

Рисунок 6. — кривая Хартера-Хейтуэя

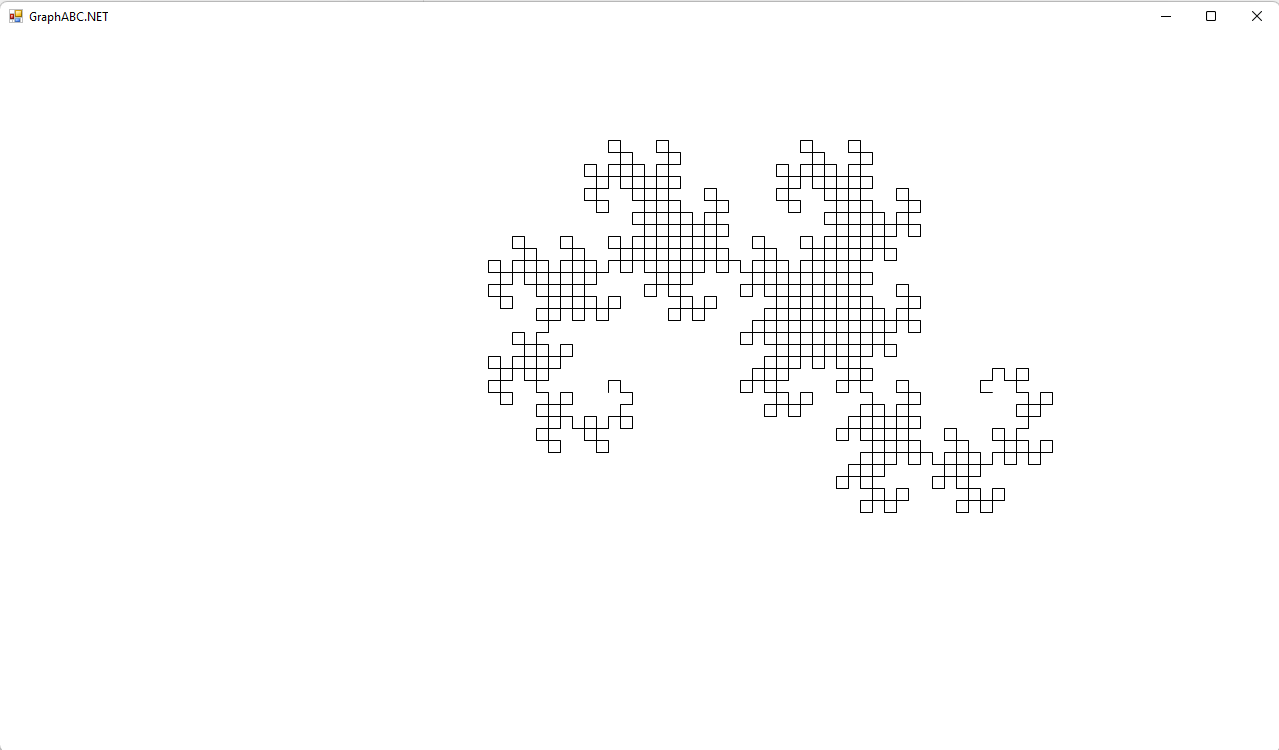


Рисунок 7. — увеличение масштаба кривой

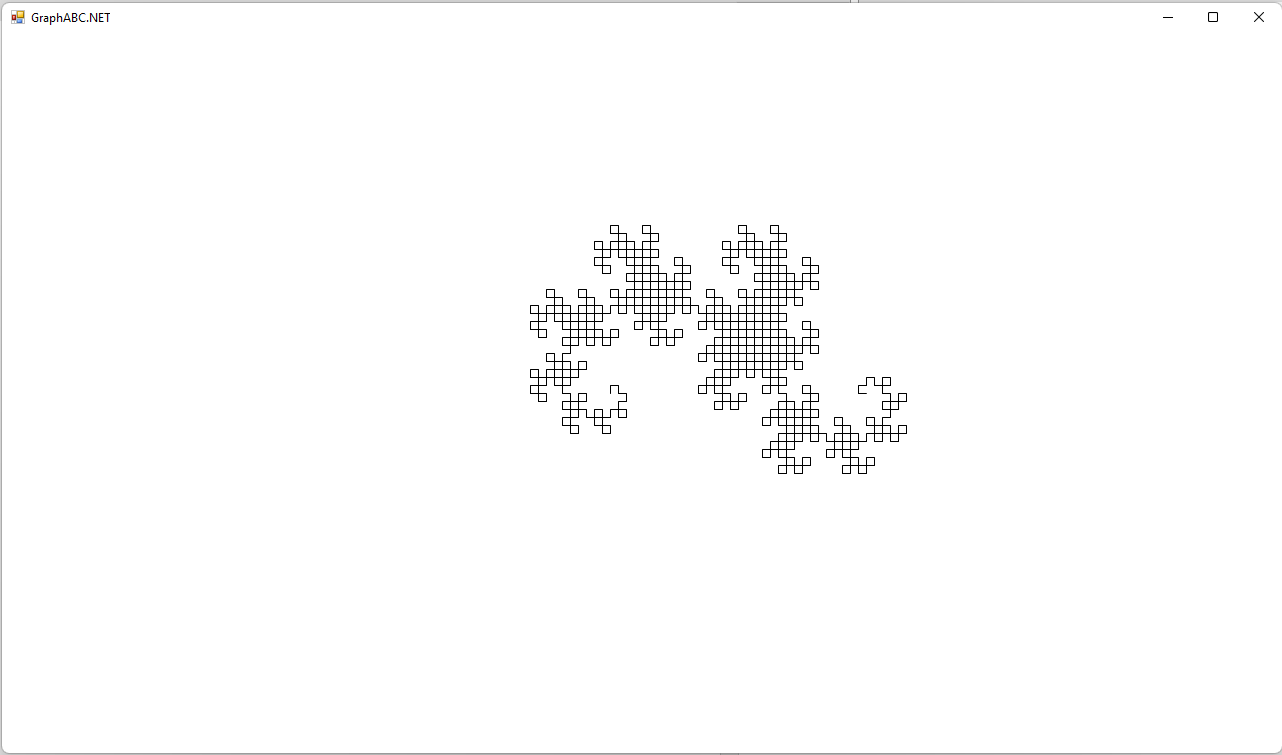


Рисунок 8. — уменьшение масштаба кривой

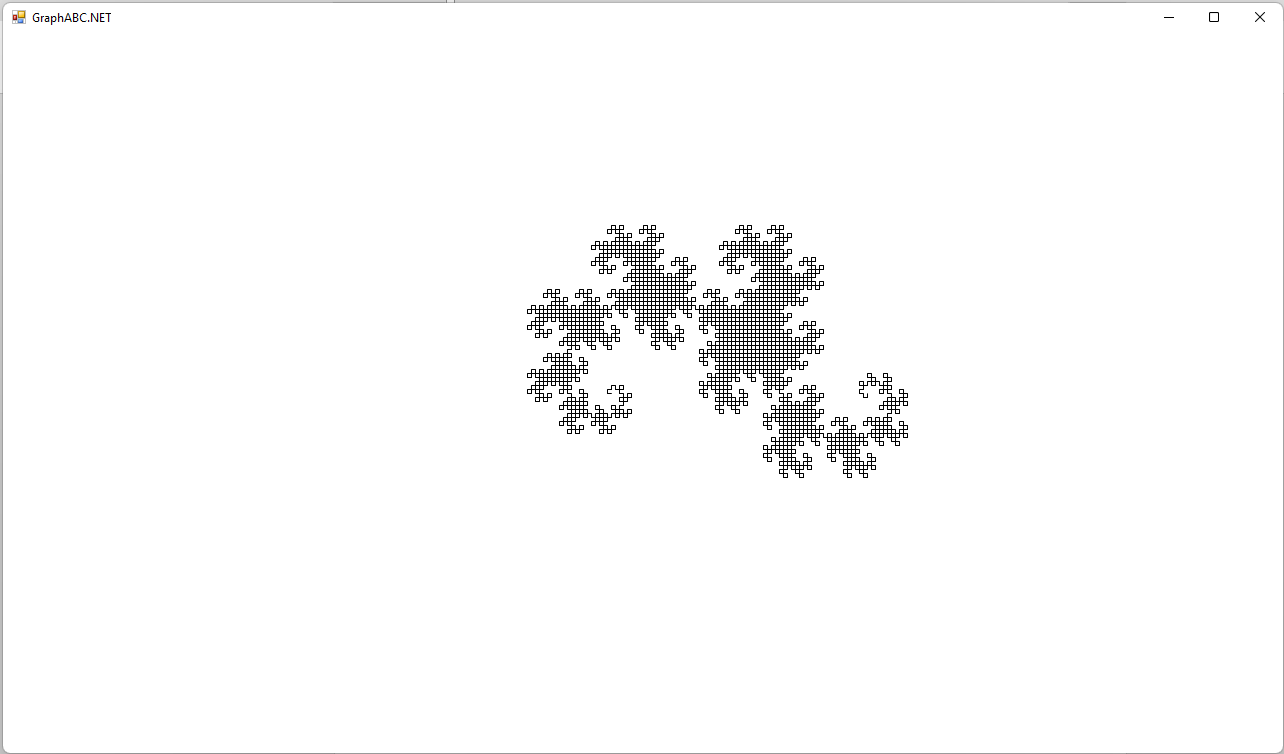


Рисунок 9. — увеличение глубины кривой

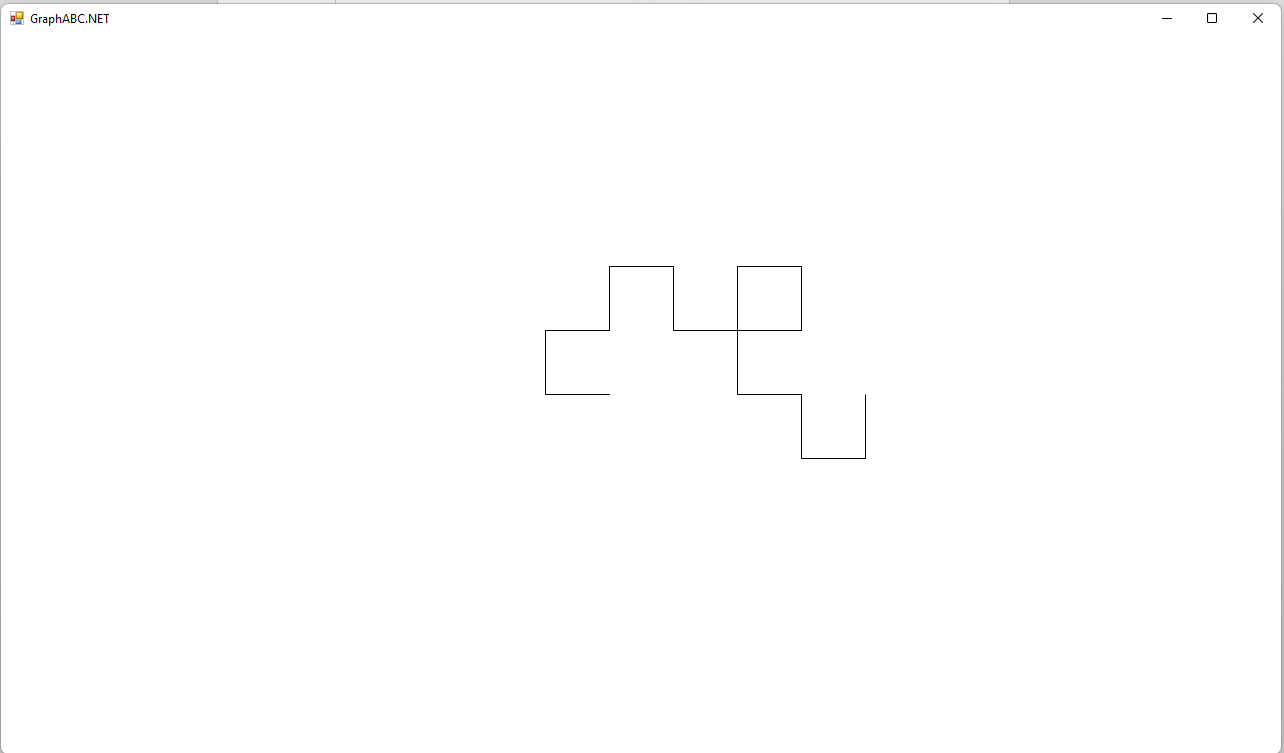


Рисунок 10. — уменьшение глубины кривой

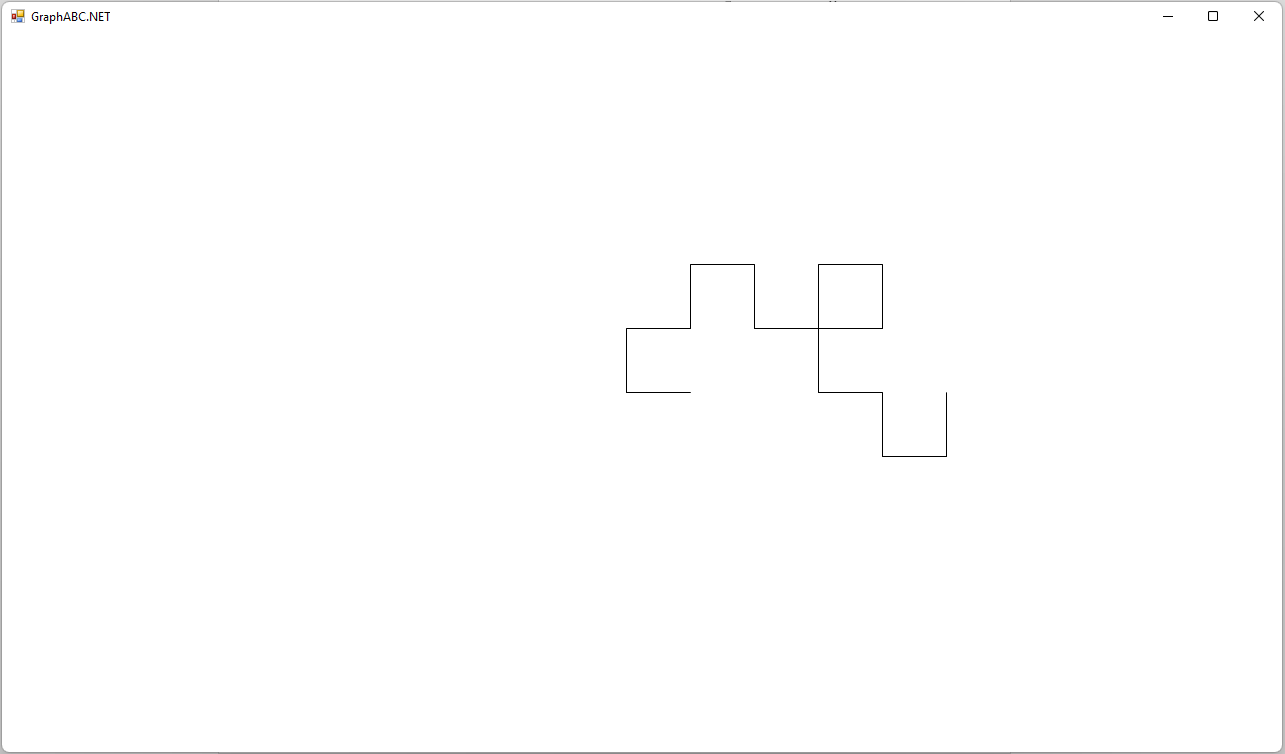


Рисунок 11. — сдвиг вправо

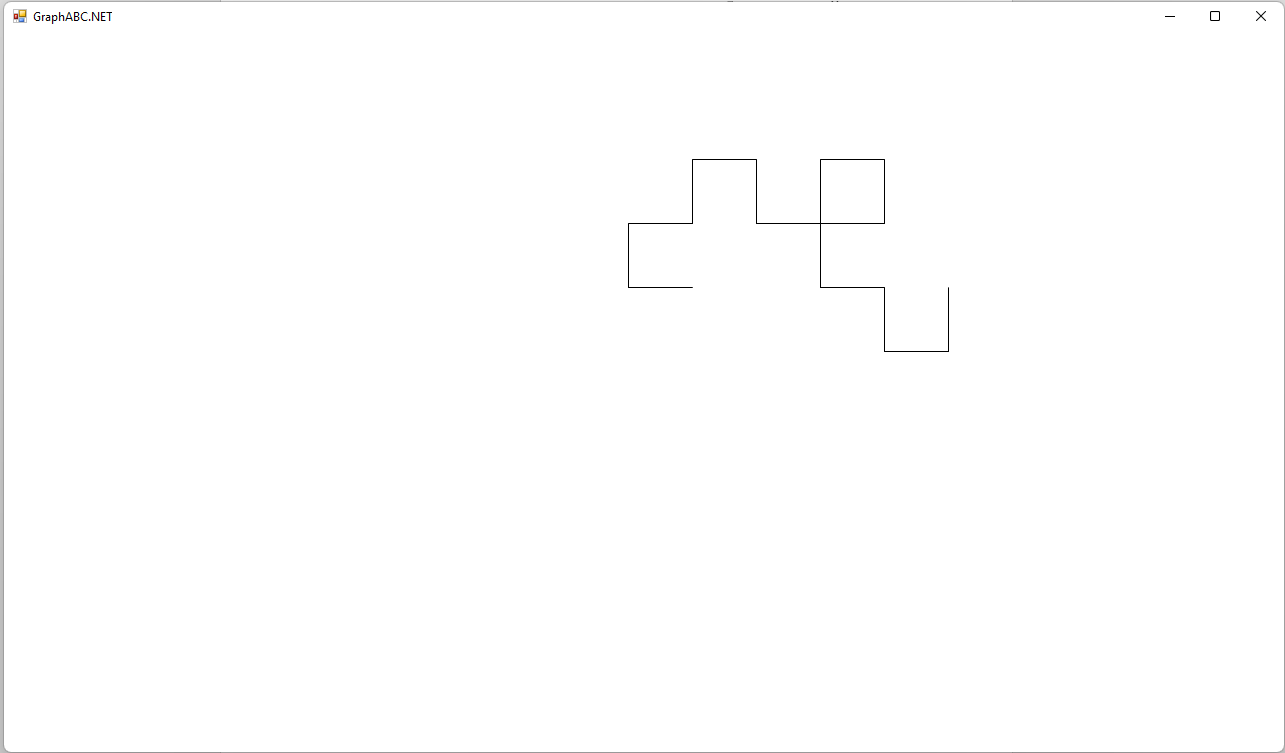


Рисунок 11. — сдвиг вверх

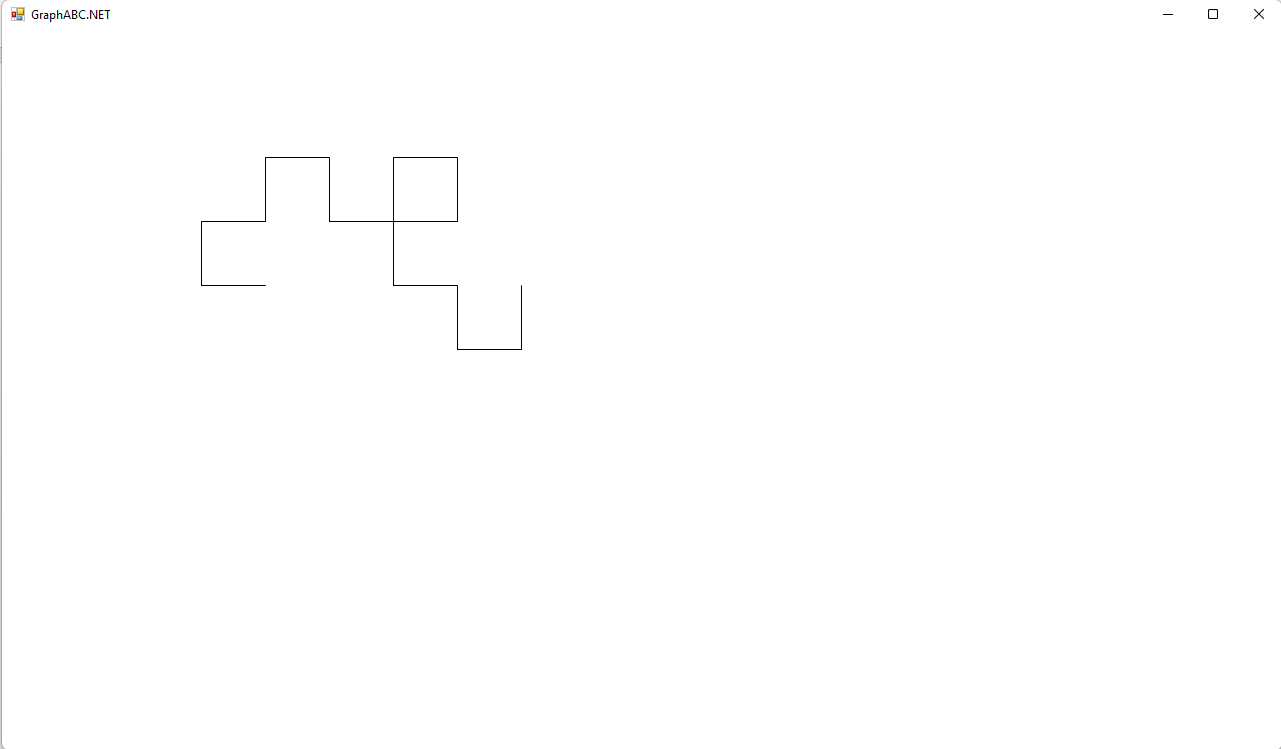


Рисунок 12. — сдвиг влево

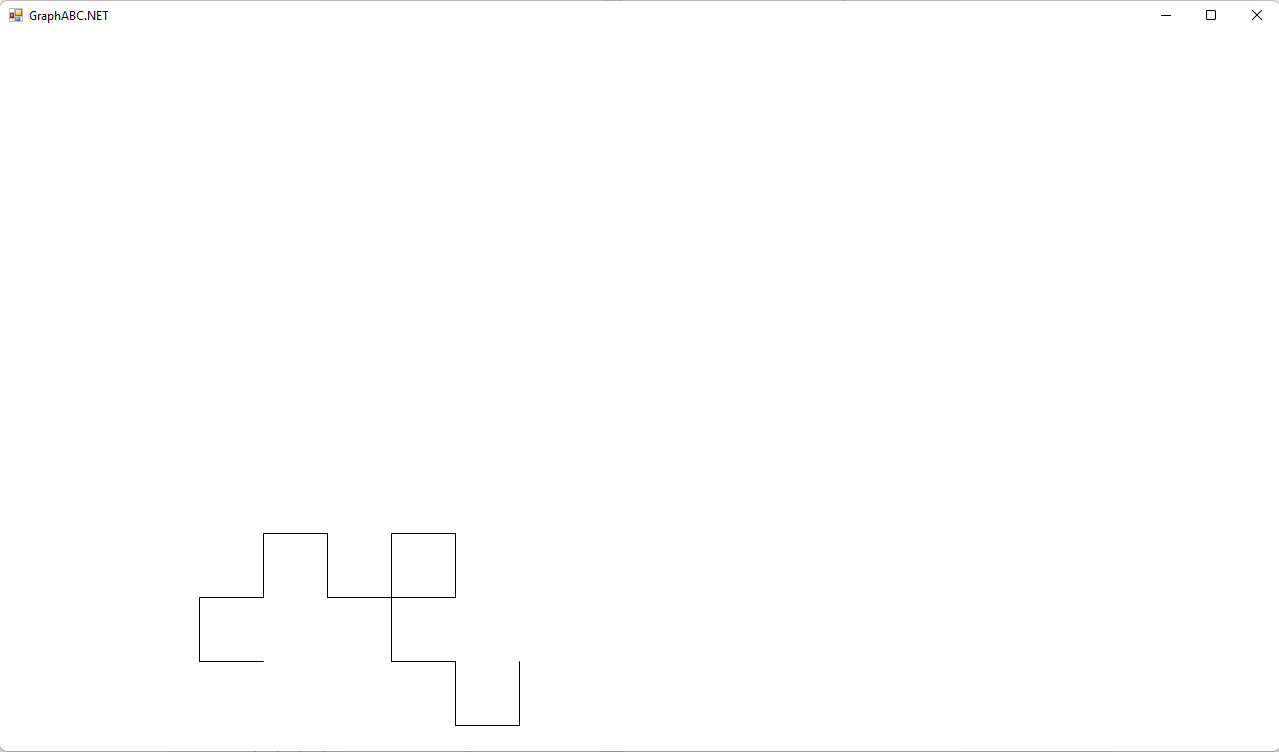


Рисунок 13. — сдвиг вниз

**Вывод**

Фракталы — это сложные структуры, имеющие свойство самоподобия, что означает, что они выглядят одинаково на различных масштабах.

Кривая Дракона — это фрактальная кривая, которая строится итеративно, начиная с прямого отрезка. Определение можно описать так:

1. Начинаем с отрезка, который мы назовем "нулевой уровень".
2. На первом шаге поворачиваем этот отрезок на 90 градусов, чтобы получить два новых отрезка.
3. На каждом следующем шаге берем последний сегмент, поворачиваем его на 90 градусов и добавляем к предыдущей фигуре.
4. Этот процесс повторяется множество раз, и в результате получается сложная, изогнутая кривая.

В ходе работы мы научились строить фракталы посредством рекурсивных процедур, а также организовывать отдельные модули, куда можно выносить отдельные процедуры и функции.