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»**

Выполнил: студент учебной группы

ИСПк-202-52-00

Порубов Кирилл Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

1. Цель контрольной работы.

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

1. Формулировка задания (с вариантом)

Вариант: 10

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

Кривая Минковского нигде не дифференцируема и не спрямляема.

Кривая Минковского не имеет самопересечений.

Кривая Минковского имеет Хаусдорфову размерность {\displaystyle \ln 8/\ln 4\ =3/2} (поскольку она состоит из восьми равных частей, каждая из которых подобна всей кривой с коэффициентом подобия 1/4). В частности, Кривая Минковского имеет нулевую меру Лебега.

Строится кривая следующим образом: изначально есть отрезок, который преобразуется в ломаную, а при каждой следующей итерации к каждому из полученных на шаге ранее звеньев ломаной применяется аналогичная процедура, что демонстрируется наглядно на рисунке ниже.

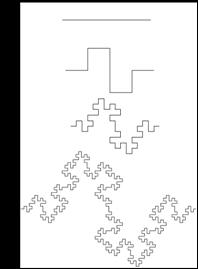


Рисунок 1 - Пример алгоритма

1. Код программы

**uses** GraphABC;

**procedure** RLine(x, y, x1, y1: real) := Line(Round(x), Round(y), Round(x1), Round(y1));

**function** GetAngle(x, y, x2, y2: real): real;

**begin**

**var** angle := Abs(RadToDeg(ArcTan((y2 - y) / (x2 - x))));

**if** (x2 = x) **and** (y2 = y) **then**

Result := 0

**else**

**if** x2 > x **then**

**if** y2 > y **then** Result := angle **else** Result := 360 - angle

**else**

**if** y2 > y **then** Result := 180 - angle **else** Result := 180 + angle;

**end**;

**function** Distance(x, y, x1, y1: real) := Sqrt(Sqr(x1 - x) + Sqr(y1 - y));

**var** m: integer;

**procedure** Draw(x, y, x1, y1: real);

**begin**

**var** r := Distance(x, y, x1, y1);

**if** r < 4\*\*m **then**

RLine(x, y, x1, y1)

**else**

**begin**

**var** angle := GetAngle(x, y, x1, y1);

**var** angleP := DegToRad(angle + 90);

**var** angleM := DegToRad(angle - 90);

r /= 4;

**var** dx := (x1 - x) / 4;

**var** dy := (y1 - y) / 4;

**var** xA := x + dx;

**var** yA := y + dy;

**var** xB := xA + dx;

**var** yB := yA + dy;

**var** xC := xB + dx;

**var** yC := yB + dy;

**var** x2 := xA + r \* Cos(angleP);

**var** y2 := yA + r \* Sin(angleP);

**var** x3 := xB + r \* Cos(angleP);

**var** y3 := yB + r \* Sin(angleP);

**var** x4 := xB + r \* Cos(angleM);

**var** y4 := yB + r \* Sin(angleM);

**var** x5 := xC + r \* Cos(angleM);

**var** y5 := yC + r \* Sin(angleM);

Draw(x, y, xA, yA);

Draw(xA, yA, x2, y2);

Draw(x2, y2, x3, y3);

Draw(x3, y3, xB, yB);

Draw(xB, yB, x4, y4);

Draw(x4, y4, x5, y5);

Draw(x5, y5, xC, yC);

Draw(xC, yC, x1, y1);

**end**;

**end**;

**var** x, y, x1, y1, k: integer;

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

**begin**

**case** key **of**

VK\_Up: **begin** y := y - 5; y1 := y1 - 5 **end**;

VK\_Down: **begin** y += 5; y1 += 5 **end**;

VK\_Left: **begin** x := x - 5; x1 := x1 - 5 **end**;

VK\_Right: **begin** x := x + 5; x1 := x1 + 5 **end**;

VK\_A: x := x - 50;

Vk\_Z: x := x + 50;

vk\_s: **if** m>0 **then** m -= 1;

vk\_x:**if** m<4 **then** m += 1;

vk\_F:

**end**;

Window.Clear;

draw(x, y, x1, y1);

redraw;

**end**;

**begin**

LockDrawing;

x := 100;

y := 200;

x1 := 400;

y1 := 200;

m:= 2;

draw(x, y, x1, y1);

redraw;

onKeyDown += keydown;

**end**.

1. Результат выполнения программы

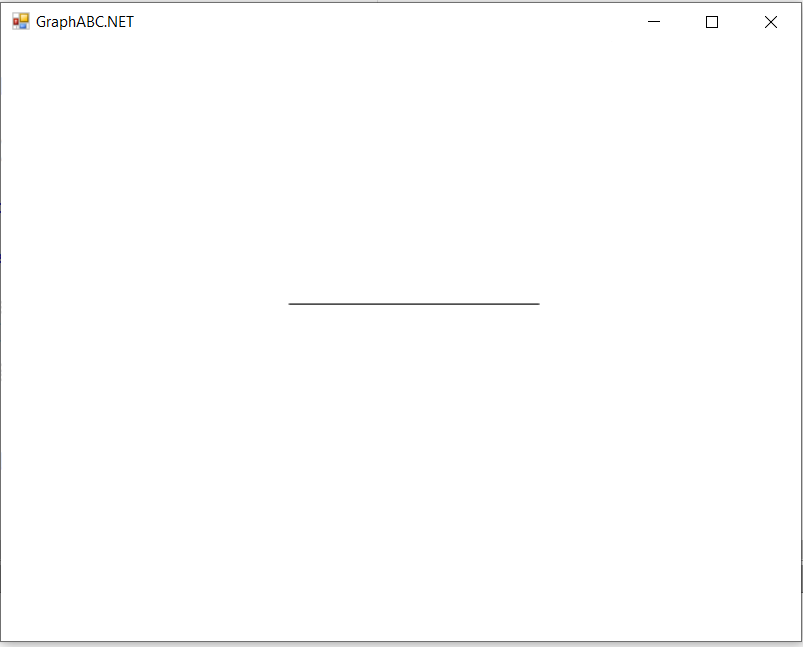


Рисунок 2 - Результат выполнения программы (1)

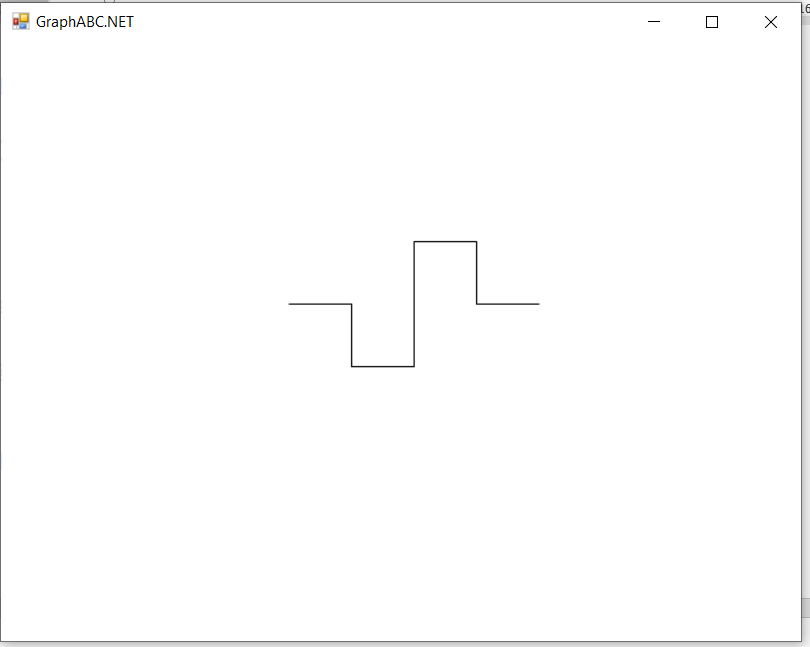


Рисунок 3 - Результат выполнения программы (2)

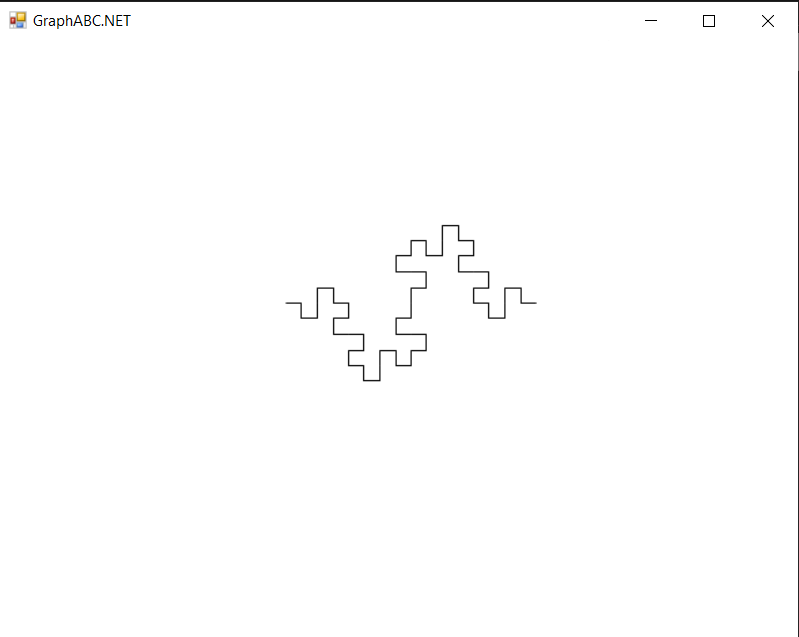


Рисунок 4 - Результат выполнения программы (3)

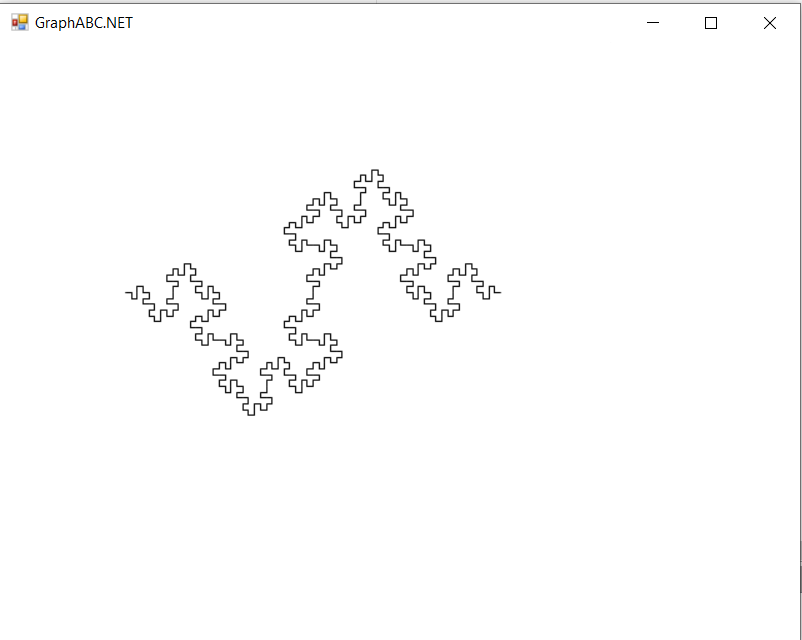


Рисунок 5 - Результат выполнения программы (4)

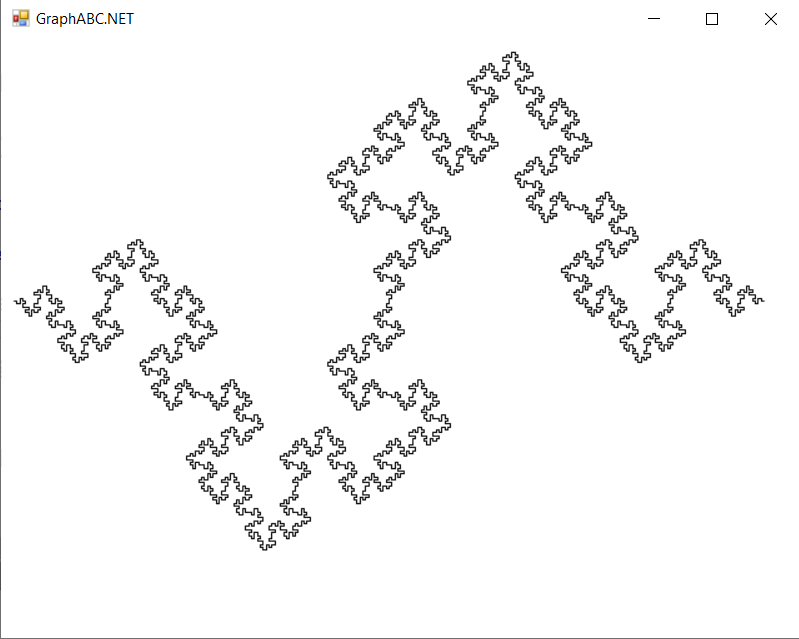


Рисунок 6 - Результат выполнения программы (5)

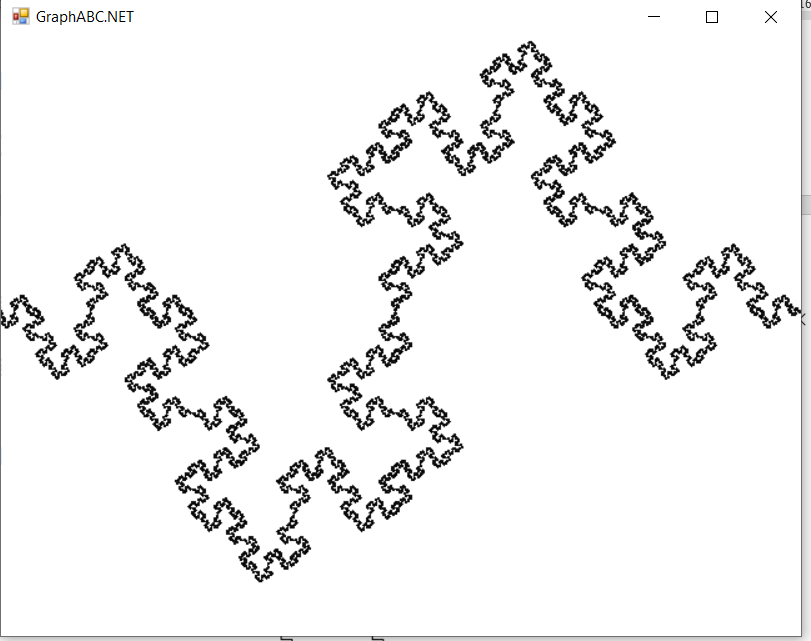


Рисунок 7 - Результат выполнения программы (6)

Вывод

В ходе выполнения домашней контрольной работы №7 «исследование фракталов» Получили навыки реализации алгоритмов с рекурсивными вычислениями, познакомились с фракталами.

В программе использовались такие функии как: Функция Draw- вывод иллюстрации, находящейся в bmp – файле; Cos: функция - Возвращается косинус параметра (X - угол в радианах); Sin: Возвращает синус параметра.

С работой справились в полном объёме. Конечно возникали трудности, однако, их удалось решить. Получили новые знания, умения, опыт. В работе помогали электронные ресурсы, знания, полученные во время занятий благодаря преподавателю и знания, полученные в процессе саморазвития.