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**

**«Фракталы и фрактальная графика»**

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

Выполнил: студент учебной группы

ИСПк-205-52-00

Арбузов Иван Андреевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**Формулировка задания:**

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.

**Описание алгоритма:**

Алгоритм построения фрактала Кривая Минковского следующий:

1. Задаем начальный набор точек, образующих начальную ломаную. Этот набор точек может быть произвольным.
2. Для каждой точки в начальной ломаной выполняем следующие операции:

* Вычисляем длину отрезка между текущей точкой и следующей точкой в ломаной.
* Вычисляем угол между отрезком и осью Ox (горизонтальной осью).
* Вычисляем новые точки, образующие два отрезка под углом 90 градусов с текущим отрезком. Длины этих отрезков равны 1/3 от исходной длины.
* Поворачиваем и смещаем новые точки относительно текущей точки и текущего отрезка.

1. Повторяем шаг 2 для каждой точки в каждой полученной ломаной, пока не достигнем нужной глубины прорисовки (задается пользователем).
2. В результате получаем множество ломанных, образующих фрактал Кривая Минковского.
3. Для визуализации фрактала используем графический интерфейс, который позволяет масштабировать, изменять глубину прорисовки и перемещать полученную фигуру.

Алгоритм реализуется в отдельном модуле, который вы можете написать самостоятельно или воспользоваться уже готовыми реализациями алгоритма Кривой Минковского на языке программирования, который вы используете.

**Схема алгоритма:**

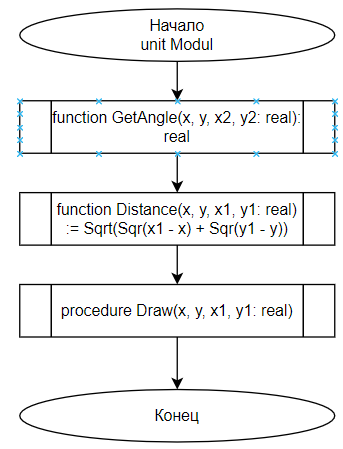


Рисунок 1 – соединение функций и процедур

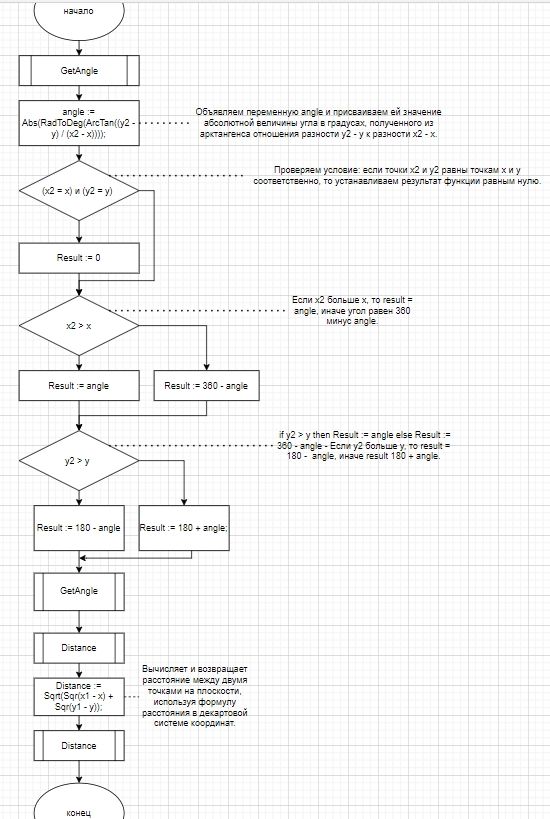


Рисунок 2

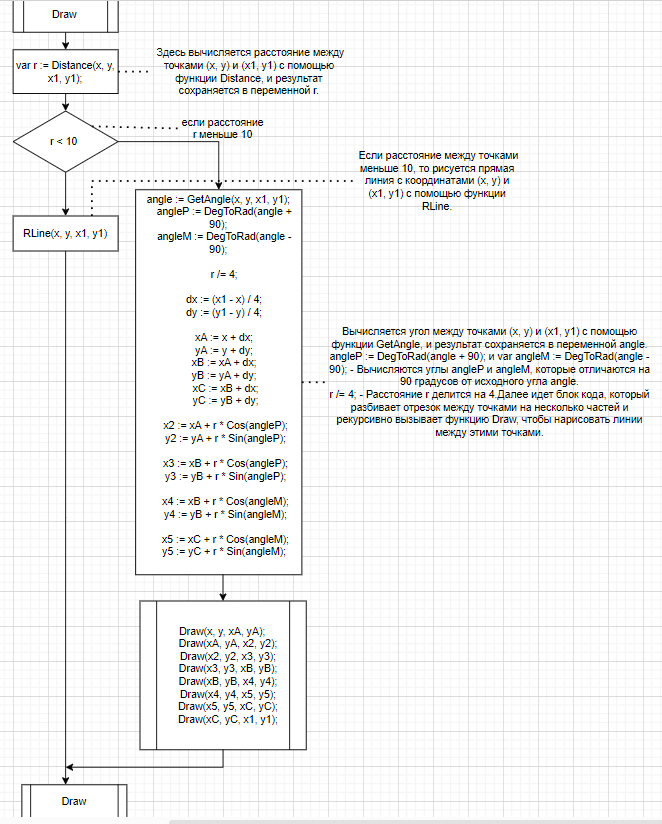


Рисунок 3

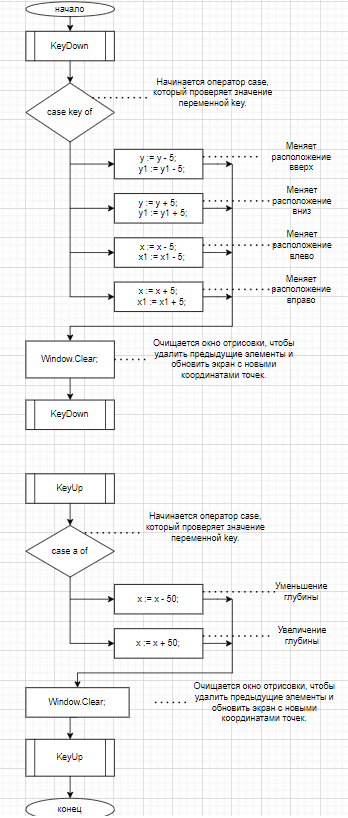
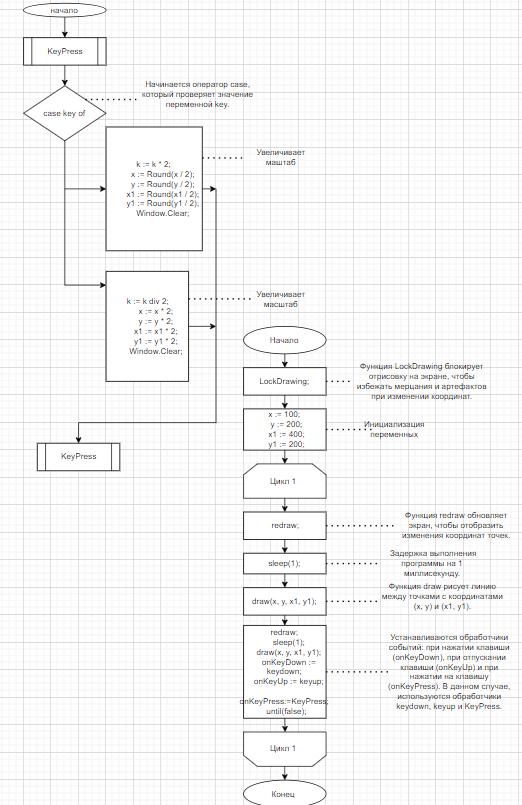


Рисунок 3



конец

Рисунок 4

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));

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

**begin**

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

**if** r < 10 **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);//Движ

**begin**

**case** key **of**

VK\_Up:

**begin**

y := y - 5;

y1 := y1 - 5;

**end**;

VK\_Down:

**begin**

y := y + 5;

y1 := y1 + 5;

**end**;

VK\_Left:

**begin**

x := x - 5;

x1 := x1 - 5;

**end**;

VK\_Right:

**begin**

x := x + 5;

x1 := x1 + 5;

**end**;

**end**;

Window.Clear;

**end**;

**procedure** KeyUp(a: integer);//глубина

**begin**

**case** a **of**

VK\_A:

**begin**

x := x - 10;

**end**;

Vk\_D:

**begin**

x := x + 10;

**end**;

vk\_s:

**begin**

x := x + 10;

**end**;

vk\_w:

**begin**

x := x + 10;

**end**;

**end**;

Window.Clear;

**end**;

**procedure** KeyPress(i: char);//масшатаб

**var** s:integer;

k:=1;

**begin**

**case** i **of**

'z':

**begin**

k := k \* 2;

**end**;

'x':

**begin**

k := k **div** 2;

**end**;

**end**;

**for** s := 1 **to** 5 **do**

**begin**

x := x \* k;

x1 := x1 \* k;

y1 := y1 \* k;

y := y \* k;

**end**;

**end**;

**begin**

LockDrawing;

x := 100;

y := 200;

x1 := 400;

y1 := 200;

**repeat**

redraw;

sleep(1);

draw(x, y, x1, y1);

onKeyDown := keydown;

onKeyUp := keyup;

onKeyPress:=KeyPress;

**until**(false);

**end**.

Результаты выполнения работы:

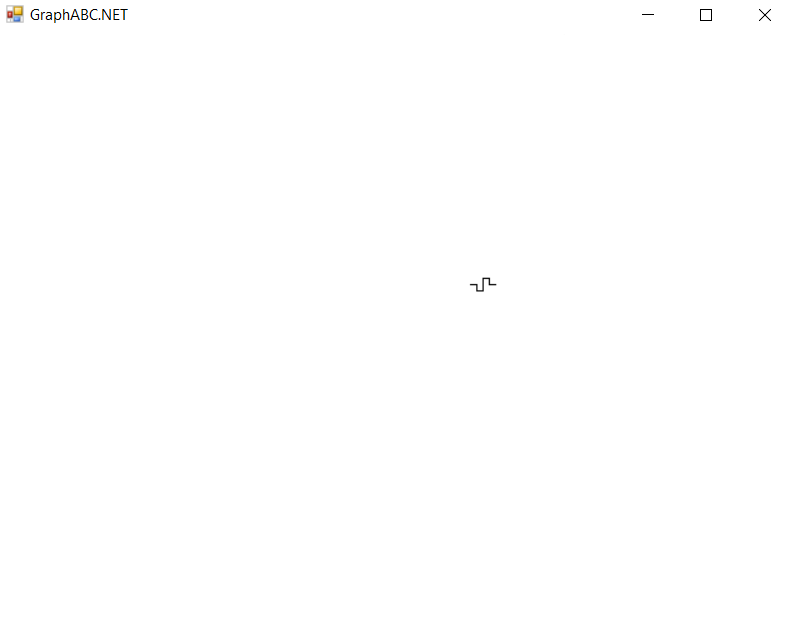


Рисунок 5



Рисунок 6

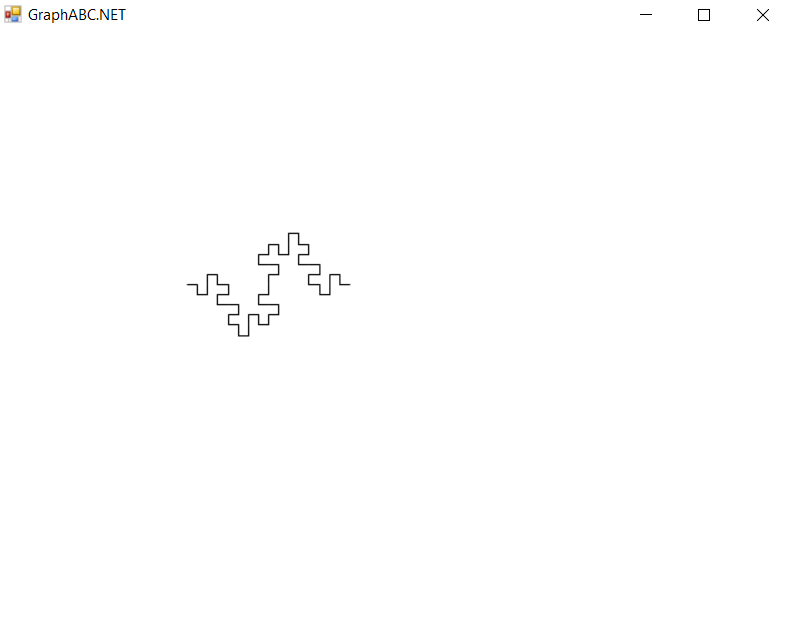
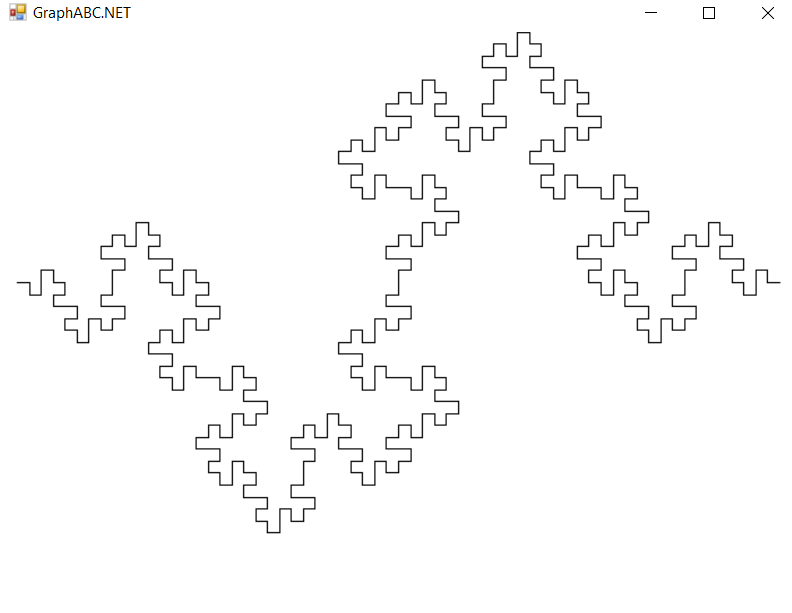


Рисунок 7

Рисунок 8

Вывод: В ходе данной домашней контрольной работы было изучены принципы работы с рекурсивными вычислениями и фракталами. Мы успешно справились с выполнением задания и были заинтересованы и увлечены процессом.