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 МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-203-52-00

Дворников Владислав Максимвоич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. Цель работы

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

1. Скриншот задания (с вариантом)

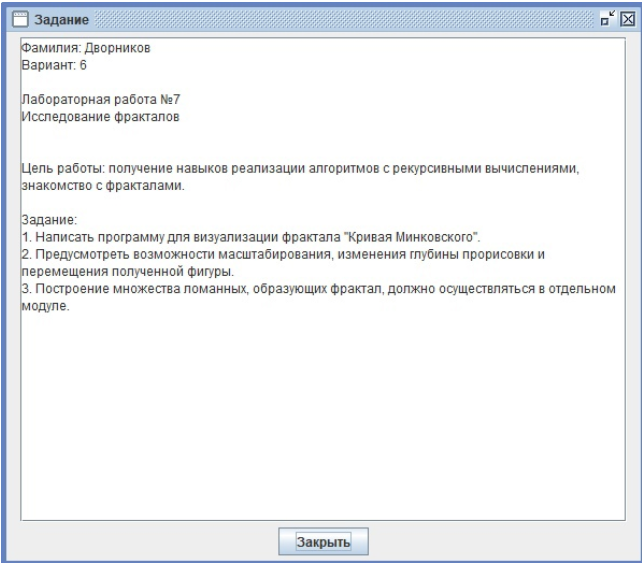


Рисунок 1 – Скриншот задания

1. Описание алгоритма

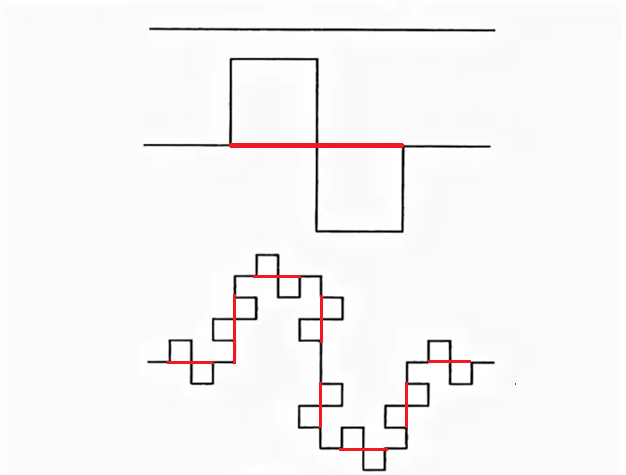


Рисунок 2 – Описание алгоритма

1. Код программы

Основная программа:

**uses** GraphABC, fractal;

**begin**

LockDrawing;

x := 100;

y := 200;

x1 := 400;

y1 := 200;

m := 2;

draw(x, y, x1, y1);

redraw;

onKeyDown += keydown;

**end**.

Модуль:

**unit** fractal;

**interface**

**uses** GraphABC;

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

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

**function** dist(x, y, x1, y1: real): real;

**var**

m: integer;

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

**var**

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

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

**Implementation**

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

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

**begin**

**var** ug := 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 := ug **else** Result := 360 - ug

**else**

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

**end**;

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

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

**begin**

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

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

line(x, y, x1, y1)

**else**

**begin**

**var** ug := ugol(x, y, x1, y1);

**var** uga := DegToRad(ug + 90);

**var** ugb := DegToRad(ug - 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(uga);

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

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

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

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

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

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

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

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**;

**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\_Add: x := x - 50;

VK\_Subtract: x := x + 50;

VK\_PageUp: **if** m > 0 **then** m -= 1;

VK\_PageDown: **if** m < 4 **then** m += 1;

**end**;

Window.Clear;

draw(x, y, x1, y1);

redraw;

**end**;

**end**.

1. Результат выполнения программы

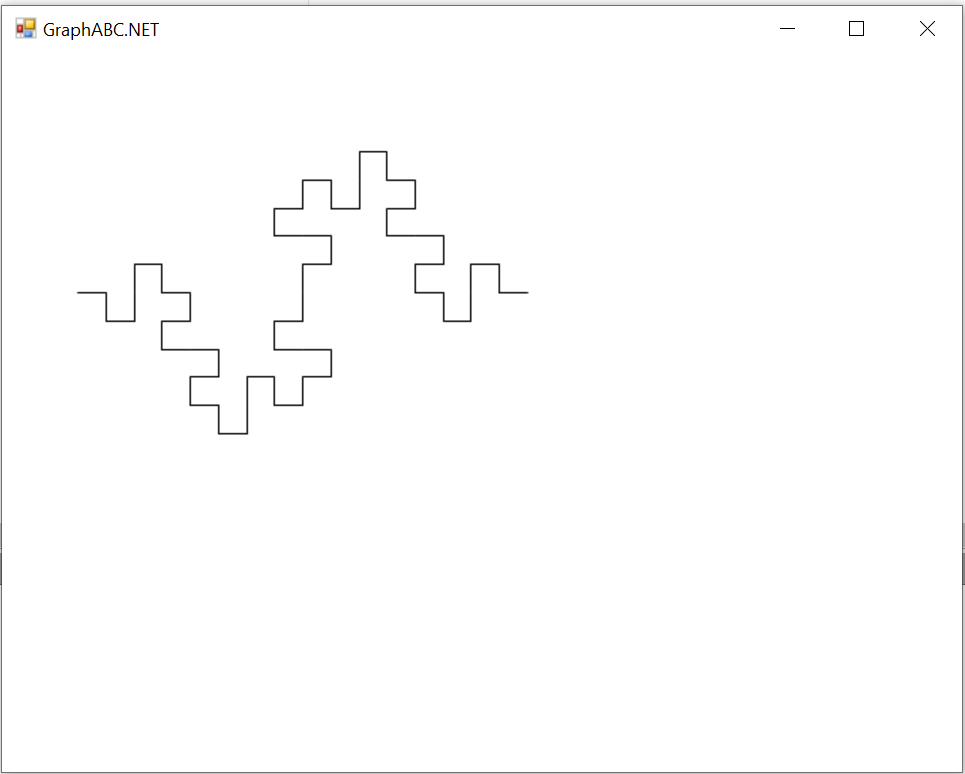


Рисунок 3 – Результат выполнения программы (1/2)

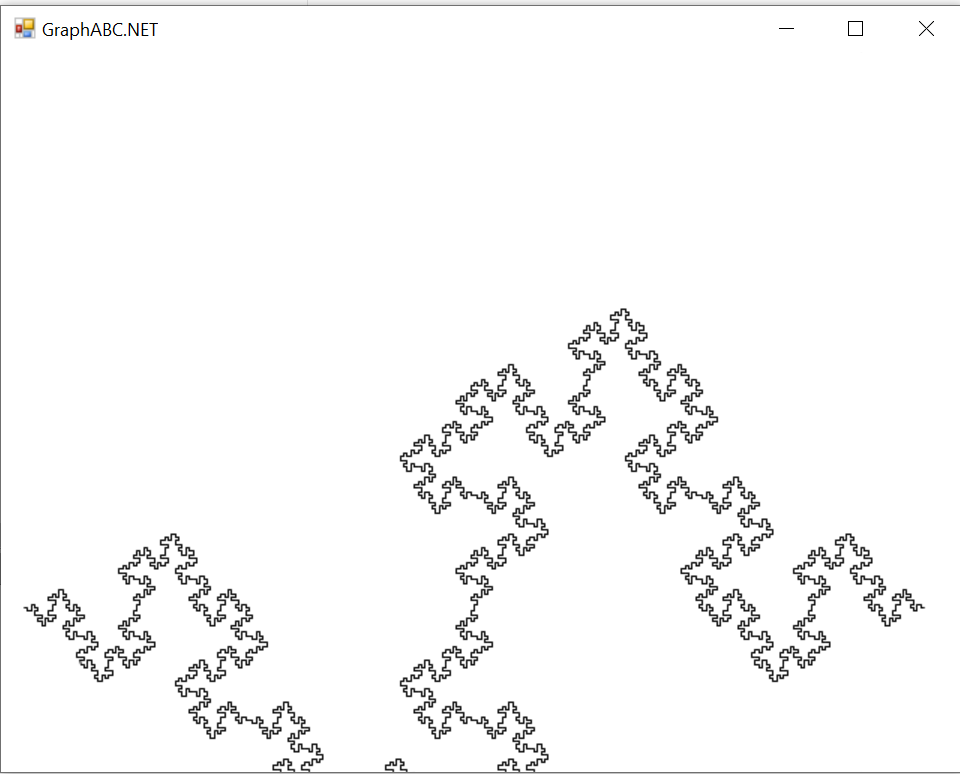


Рисунок 4 – Результат выполнения программы со сдвигом, изменённым масштабированием и изменённой глубиной прорисовки (2/2)

1. Вывод

В ходе данной работы возникло множество различных трудностей, было нелегко понять, каким образом должна быть написана программа, реализующая кривую Минковского, после долгих размышлений удалось написать данную программу, но затем было не ясно, как добавить возможность перемещения фигуры на кнопки. В конце концов, такая возможность была осуществлена с помощью использования подпрограммы с виртуальными клавишами в языке PascalABC. В конечном итоге программа была реализована в полном размере и все проблемы были решены.