

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Отчет по лабораторной работе №7
по дисциплине
«Программирование»

Выполнил студент гр. ИВТб-1301-05-00	_____ /Макаров С.А./
Руководитель зав. кафедры ЭВМ	_____ /Долженкова М.Л./

Киров 2025

Цель

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

Задание

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.

Решение

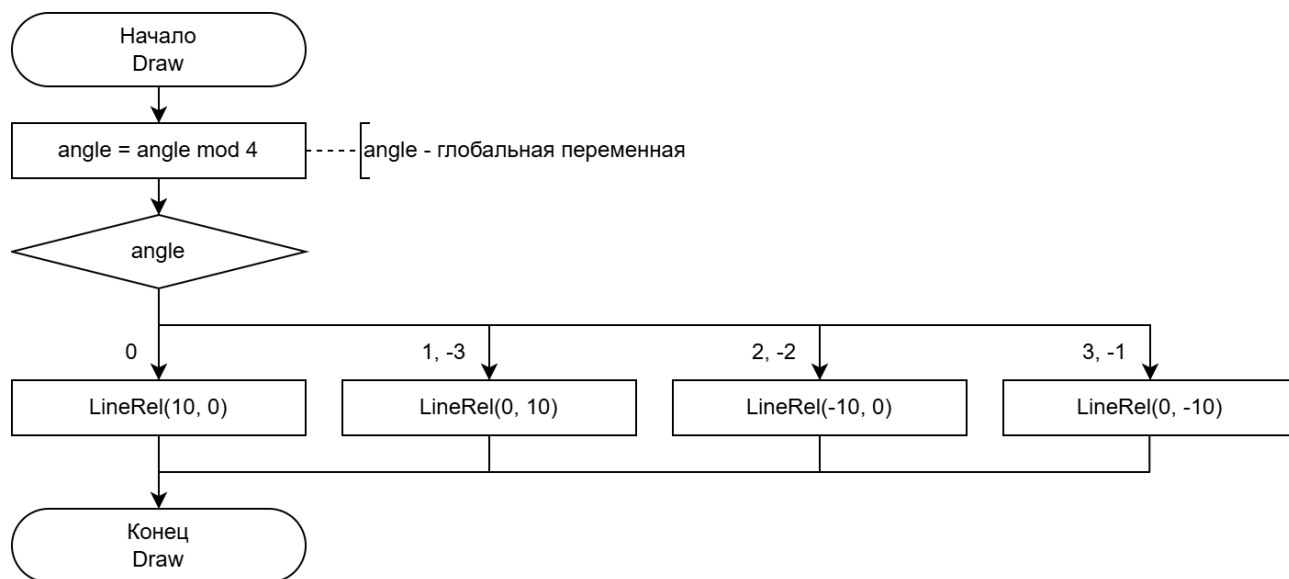


Рисунок 1 – Схема алгоритма отрисовки отрезка

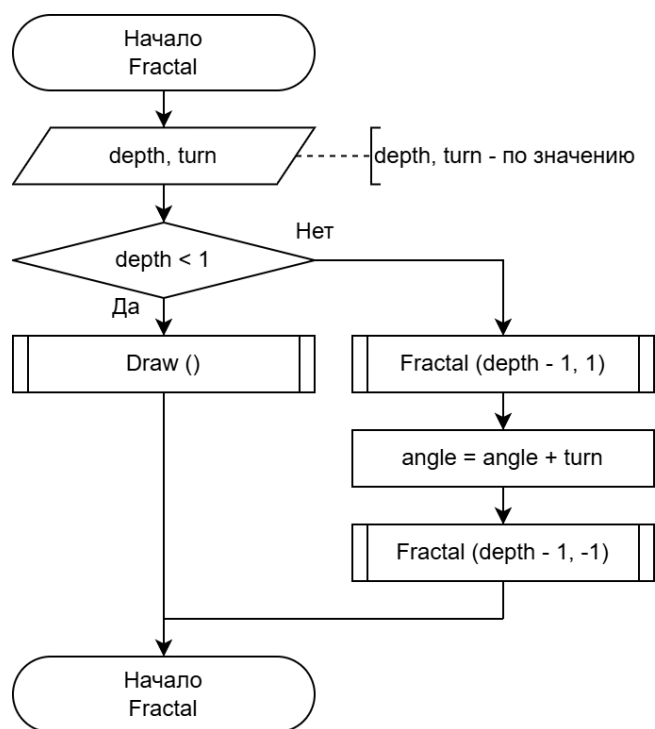


Рисунок 2 – Схема алгоритма построения кривой

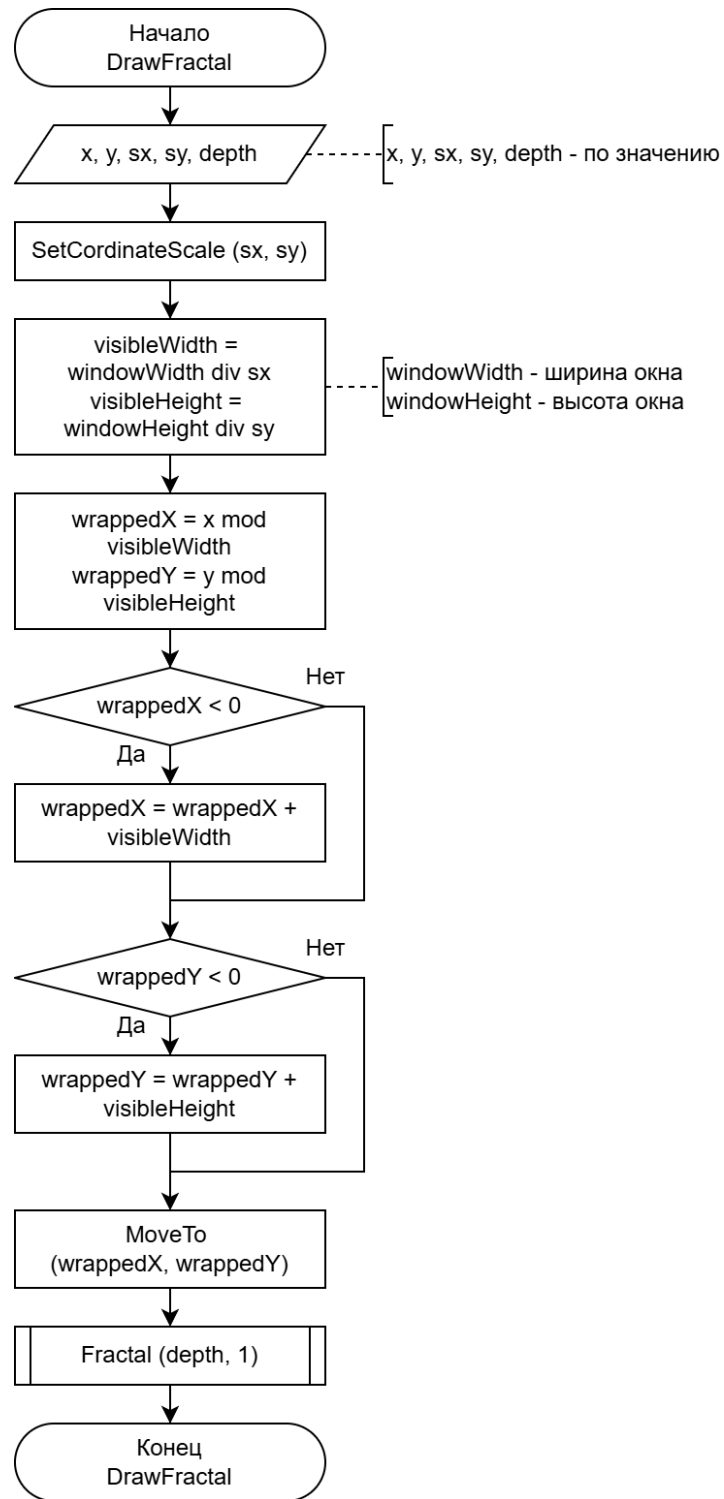


Рисунок 3 – Схема алгоритма построения фрактала

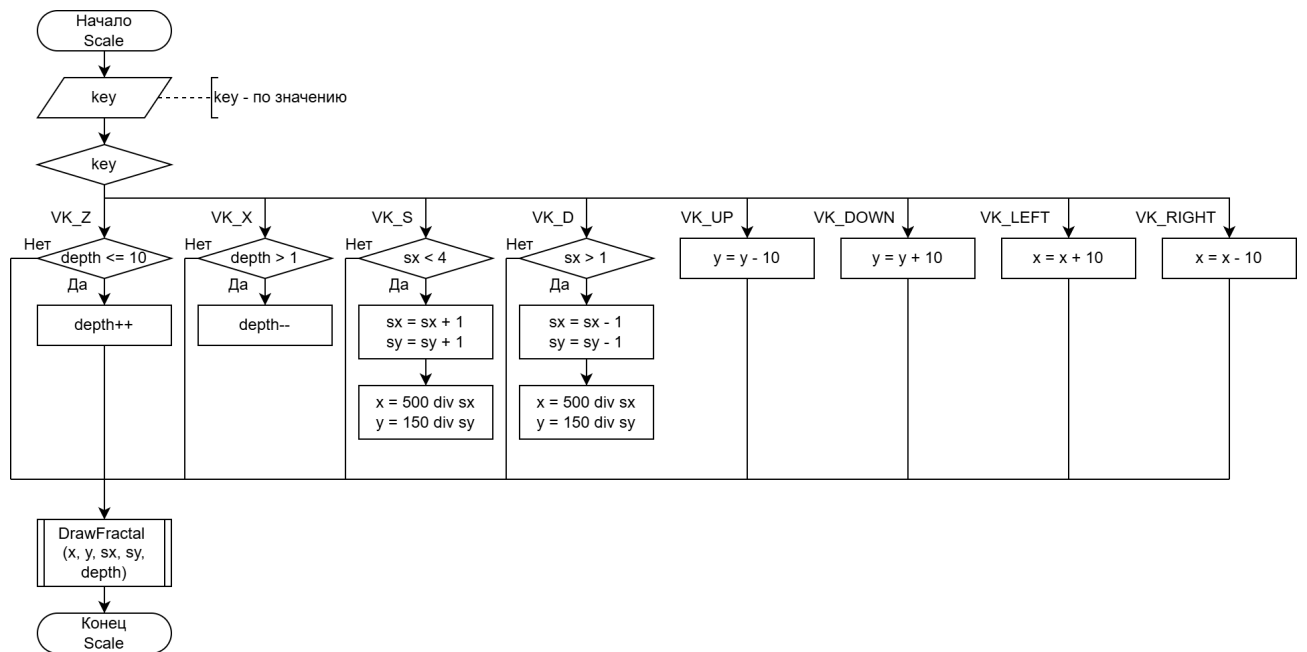


Рисунок 4 – Схема алгоритма обработки клавиш

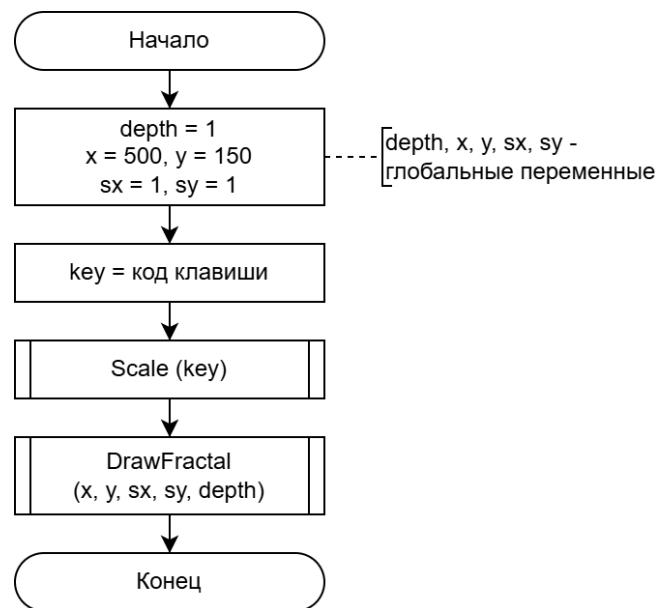


Рисунок 5 – Схема алгоритма программы

Исходный код модуля для построения множества ломанных:

```
unit Dragon;
interface
uses
    GraphABC;
procedure DrawFractal(x, y, sx, sy, depth: integer);
implementation
    var
        angle: integer;
    procedure Draw();
    begin
        angle := angle mod 4;
        case angle of
            0: LineRel(10, 0);
            1, -3: LineRel(0, 10);
            2, -2: LineRel(-10, 0);
            3, -1: LineRel(0, -10);
        end;
    end;
    end;
    procedure Fractal(depth, turn: integer);
    begin
        if depth < 1 then
            begin
                Draw();
                exit;
            end;
            Fractal(depth - 1, 1);
            angle := angle + turn;
            Fractal(depth - 1, -1);
        end;
    procedure DrawFractal(x, y, sx, sy, depth: integer);
    var
        wrappedX, wrappedY: integer;
        visibleWidth, visibleHeight: integer;
    begin
        SetCoordinateScale(sx, sy);
        visibleWidth := WindowWidth div sx;
        visibleHeight := WindowHeight div sy;
        wrappedX := x mod visibleWidth;
        wrappedY := y mod visibleHeight;
```

```

    if wrappedX < 0 then wrappedX := wrappedX + visibleWidth;
    if wrappedY < 0 then wrappedY := wrappedY + visibleHeight;
    MoveTo(wrappedX, wrappedY);
    angle := 0;
    Fractal(depth, 1);
end;
end.

```

Исходный код основного модуля представлен ниже:

```

uses
    GraphABC, Dragon;
var
    depth: integer = 1;
    x: integer = 500;
    y: integer = 150;
    sx: integer = 1;
    sy: integer = 1;
procedure Scale(key: integer);
begin
    case key of
        VK_Z:
            begin
                if depth <= 10 then
                    depth += 1
                else
                    exit;
                end;
        VK_X:
            begin
                if depth > 1 then
                    depth -= 1
                else
                    exit;
                end;
        VK_S:
            if sx < 4 then
                begin
                    sx := sx + 1;
                    sy := sy + 1;
                    x := 500 div sx; y := 150 div sy;

```

```

        end;
VK_D:
    if sx > 1 then
    begin
        sx := sx - 1;
        sy := sy - 1;
        x := 500 div sx; y := 150 div sy;
    end;
VK_UP:
    y := y - 10;
VK_DOWN:
    y := y + 10;
VK_LEFT:
    x := x - 10;
VK_RIGHT:
    x := x + 10;
end;
Window.Clear();
DrawFractal(x, y, sx, sy, depth);
end;
begin
    SetWindowSize(800, 600);
    OnKeyDown := Scale;
    DrawFractal(x, y, sx, sy, depth);
end.

```

Вывод

В ходе выполнения лабораторной работы получены навыки реализации алгоритма с рекурсивным вычислением, а также был построен фрактал «Кривая Хартера-Хейтуэя».