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**

**«ИСЛЕДОВАНИЕ ФРАКТАЛА»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-203-52-00

Цветкова Кристина Андреевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

Содержание

1. Цель работы: получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.
2. Формулировка задания (с вариантом)

Вариант 22

Задание:

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

Программа начинается с входных данных (начальное кол-во итераций, начальные координаты и расстояние между линиями). После идет функция per которая отвечает за очистку окна после каждого нажатия на кнопки управления, за кнопку которую нажал пользователь и вход в цикл рисования фрактала. Далее идут функции KU, KD, KR, KL которые нужны для вызова самих себя и LineUP, LineDOWN, LineRIGHT, LineLEFT которые рисуют линии и описаны в библиотеке fractal.

1. Схема алгоритма с комментариями
2. Код программы

**uses** GraphABC, fractal;

**var**

x0, y0, x00, y00, step, iter: integer;

**function** KU(i:integer):integer; **forward**;

**function** KD(i:integer):integer; **forward**;

**function** KR(i:integer):integer; **forward**;

**function** KL(i:integer):integer; **forward**;

**function** per: integer; **forward**;

**function** KU( i: integer): integer;

**begin**

**if** i > 0 **then**

**begin**

KR(i - 1);

LineUP(x0, y0, step);

KU(i - 1);

LineRIGHT(x0, y0, step);

KU(i - 1);

LineDOWN(x0, y0, step);

KL(i - 1);

**end**;

**end**;

**function** KD( i: integer): integer;

**begin**

**if** i > 0 **then**

**begin**

KL(i - 1);

LineDOWN(x0, y0, step);

KD(i - 1);

LineLEFT(x0, y0, step);

KD(i - 1);

LineUP(x0, y0, step);

KR(i - 1);

**end**;

**end**;

**function** KL( i: integer): integer;

**begin**

**if** i > 0 **then**

**begin**

KD(i - 1);

LineLEFT(x0, y0, step);

KL(i - 1);

LineDOWN(x0, y0, step);

KL(i - 1);

LineRIGHT(x0, y0, step);

KU(i - 1);

**end**;

**end**;

**function** KR( i: integer): integer;

**begin**

**if** i > 0 **then**

**begin**

KU(i - 1);

LineRIGHT(x0, y0, step);

KR(i - 1);

LineUP(x0, y0, step);

KR(i - 1);

LineLEFT(x0, y0, step);

KD(i - 1);

**end**;

**end**;

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

**begin**

**case** key **of**

VK\_Up : **begin** y00:=y00-10; per; **end**;

VK\_Down : **begin** y00:=y00+10; per; **end**;

VK\_Left : **begin** x00:=x00-10; per; **end**;

VK\_Right : **begin** x00:=x00+10; per; **end**;

VK\_q : **begin** step:=step+2; per; **end**; //увеличение

VK\_w : **begin if** step > 0 **then** step:=step-2; per; **end**; //уменьшение

VK\_z : **begin if** iter<10 **then** iter:=iter+1; per; **end**; //итерация +1

VK\_x : **begin if** iter>0 **then** iter:=iter-1; per; **end**; //итерация -1

**end**;

**end**;

**function** per: integer;

**begin**

MaximizeWindow;

clearwindow;

x0:=x00;

y0:=y00;

LockDrawing;

KU(iter);

redraw;

onKeyDown:=control;

**end**;

**begin**

iter:=1;

x00:=20;

y00:=1000;

step:=100;

per;

**end**.

Код модуля фрактала:

**unit** fractal;

**interface**

**uses** GraphABC;

**procedure** LineUP(**var** x0, y0, step: integer);

**procedure** LineDOWN(**var** x0, y0, step: integer);

**procedure** LineRIGHT(**var** x0, y0, step: integer);

**procedure** LineLEFT(**var** x0, y0, step: integer);

**implementation**

**procedure** LineUP(**var** x0, y0, step: integer);

**begin**

line(x0, y0, x0, y0 - step);

y0 := y0 - step;

**end**;

**procedure** LineDOWN(**var** x0, y0, step: integer);

**begin**

Line(x0, y0, x0, y0 + step);

y0 := y0 + step;

**end**;

**procedure** LineRIGHT(**var** x0, y0, step: integer);

**begin**

Line(x0, y0, x0 + step, y0);

x0 := x0 + step;

**end**;

**procedure** LineLEFT(**var** x0, y0, step: integer);

**begin**

Line(x0, y0, x0 - step, y0);

x0 := x0 - step;

**end**;

**end**.

1. Результат выполнения программы

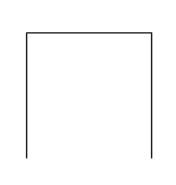
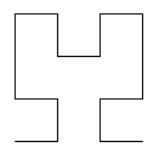
Рисунок 1 – Итерация +1

Рисунок 2 – Итерация +2

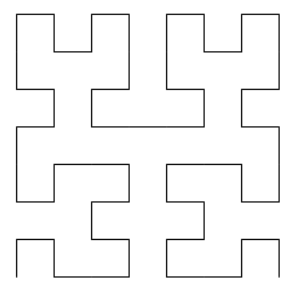


Рисунок 3 – Итерация +3

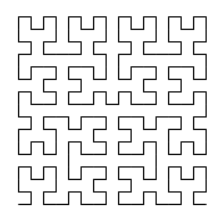


Рисунок 4 – Итерация +4

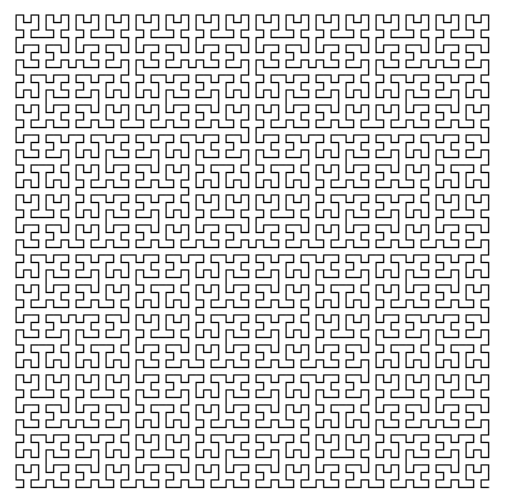


Рисунок 5 – Итерация +5

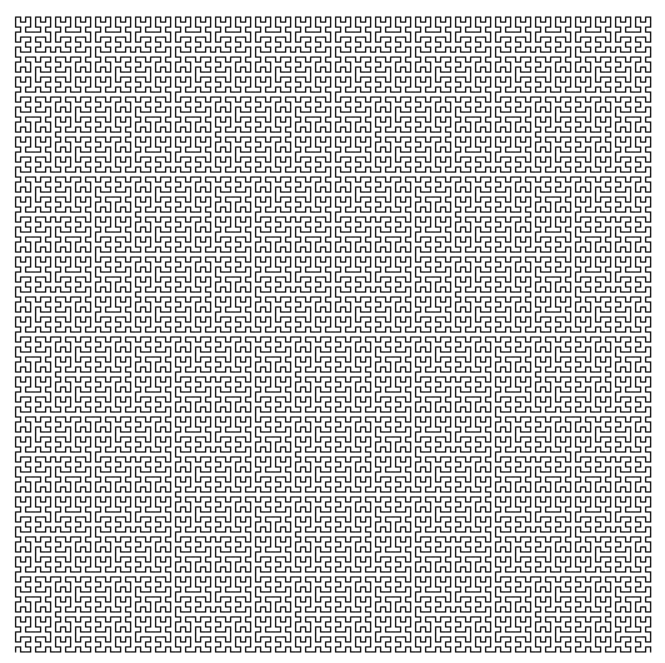


Рисунок 6 – Итерация +6

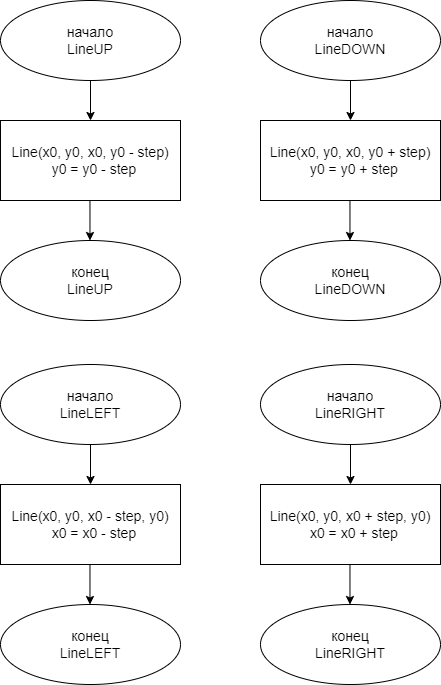
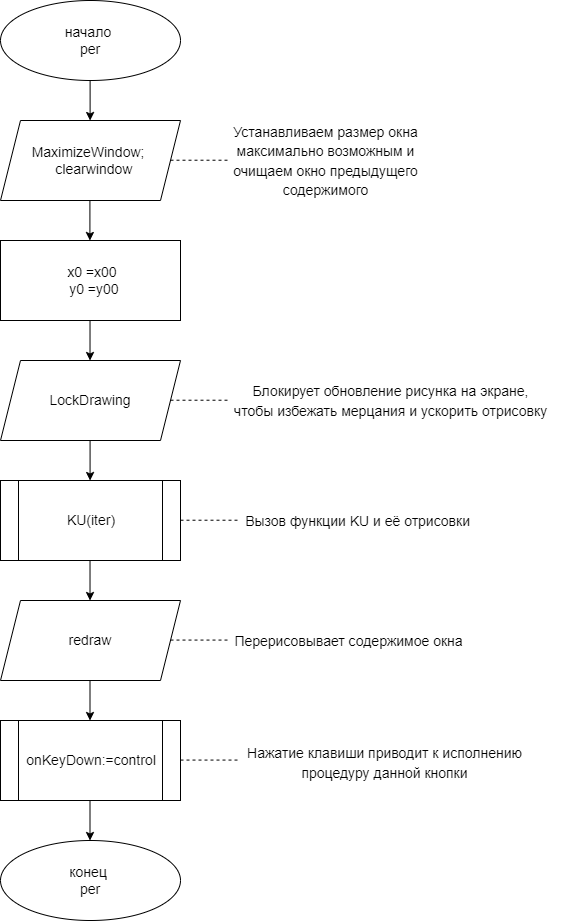
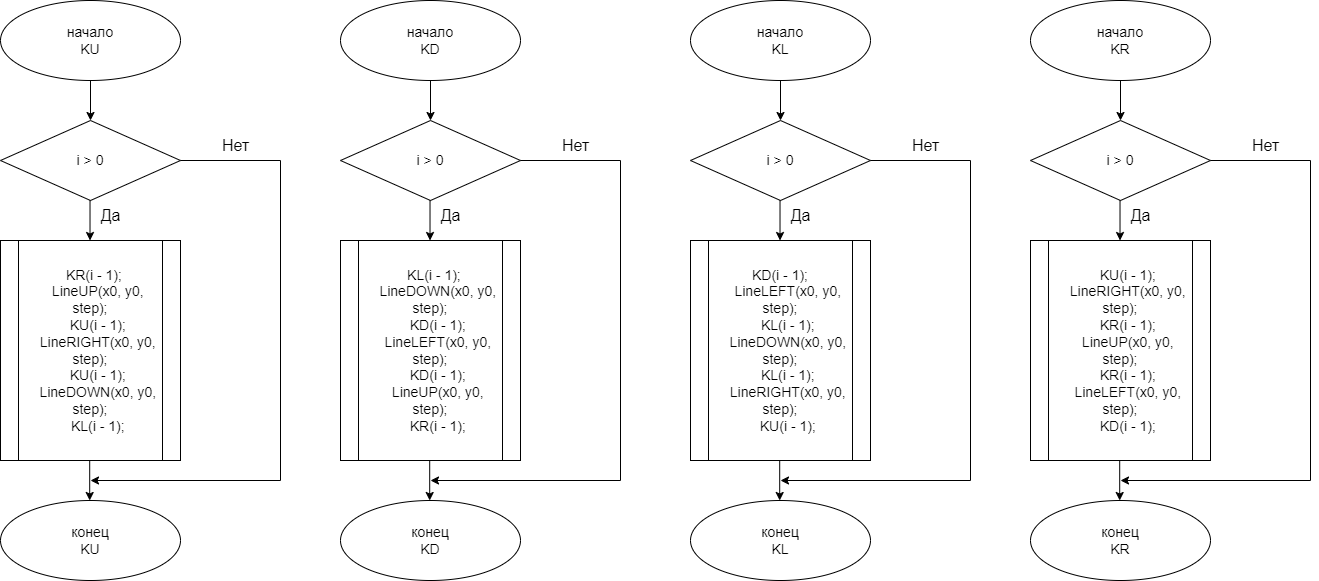


Рисунок 7 – Линии

Рисунок 8 – функция per



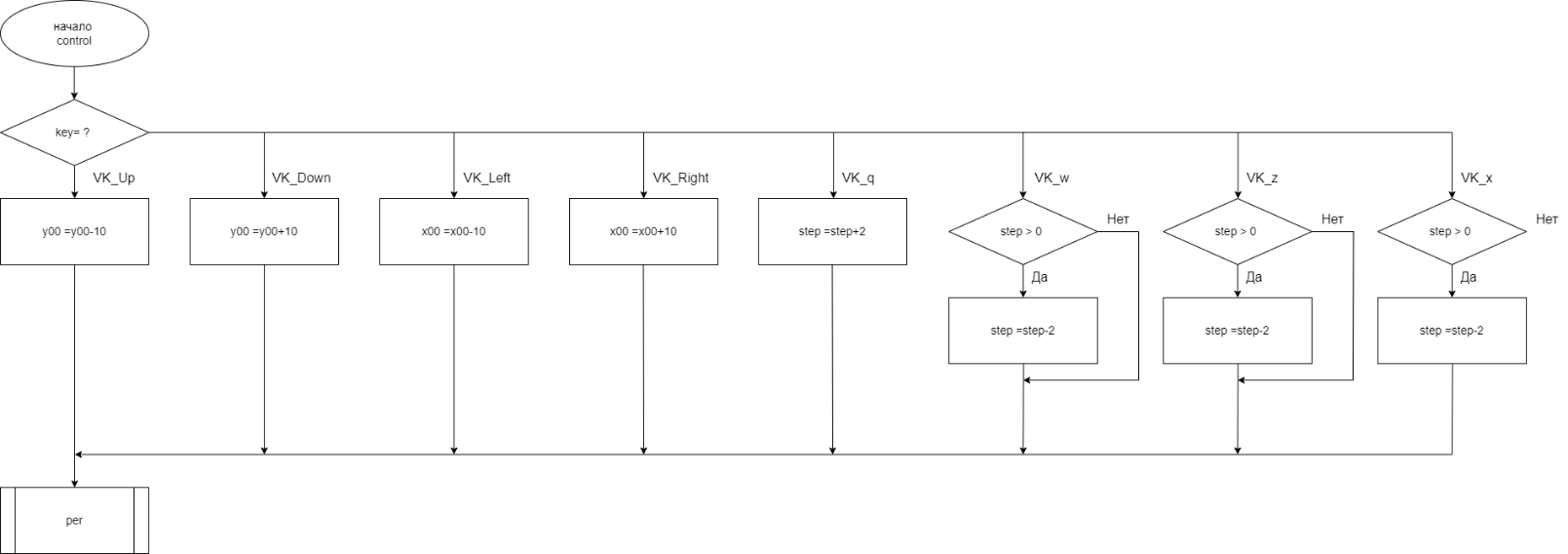
Рисунок 9 – Функции

Рисунок 10 – Меню

1. Вывод

В этой домашней контрольной работе использовалась косвенная рекурсия и была изучена кривая Пеано.

Программа начинается с того что вызывает функцию, которая меняет масштаб, количество итераций фрактала, перемещает фигуру, и вызывать функцию которая будет рисовать фигуру.

В ходе разработки кода встретились проблемы с написанием правильной последовательности рисования линий во фрактале и созданием отдельной библиотеки(модуля) в программе.

В процессе выполнения работы, для создания блок схемы мы познакомились с программой Draw io. Draw.io — инструмент для создания диаграмм, схем алгоритмов, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя.

Фрактал – это сложная, бесконечно самоподобная геометрическая фигура, каждый фрагмент которой повторяется при уменьшении масштаба.

– Геометрические;

– Алгебраические;

– Стохастические;

Фрактал «Кривая Пеано» строится на плоскости и начинается с прямой линии, которая затем разделяется на 4 части. Каждая часть заменяется на другую, которая повторяет структуру всего фрактала. Этот процесс повторяется множество раз.

Части фрактала:

Каждая из частей фрактала имеет свою форму и направление. Верхняя часть (KU) и нижняя часть (KD) состоят из прямых линий, которые поворачиваются на 90 градусов влево или вправо. Правая часть (KR) и левая часть (KL) состоят из кривых линий, которые также поворачиваются на 90 градусов.

Рекурсия предполагает вызов функцией самой себя, что позволяет повторять определенные действия или операции. В данном случае функции LineUP, LineDOWN, LineRIGHT и LineLEFT не вызывают сами себя, а выполняют определенные действия по рисованию линий в указанном направлении.

В данном коде рекурсивная триада представлена функциями KU, KD, KL, KR.

Все эти функции вызывают сами себя с измененными аргументами, что и является рекурсивной триадой.

Процедуры:

- control(key) - процедура для обработки нажатий клавиш, отвечает за управление фракталом (перемещение, масштабирование, изменение итераций)

- per - процедура для отображения фрактала, вызывается при каждом изменении параметров фрактала