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-101-51-00

Дуркин Арсений Валерьевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

В отчете должны отображаться:

1. Цель работы
2. Скриншот задания (с вариантом)
3. Описание алгоритма
4. Код программы
5. Результат выполнения программы
6. Вывод

Цель работы – получение навыков реализации алгоритмов с рекурсивными вычисленими, знакомство с фракталами.

Скриншот задания с вариантом.

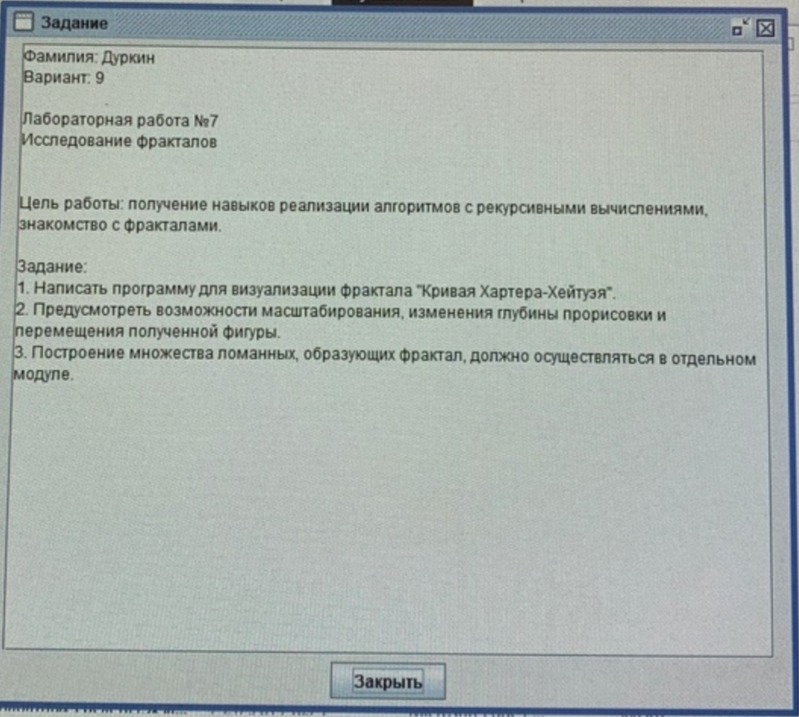


Рисунок 1 – Скриншот задания

Описание алгоритма

Берется отрезок, который сгибается пополам. Эта итерация повторяется многократно. Если согнуть это линию снова, так чтобы се углы были равны 90%, то получается драконовая ломаная.

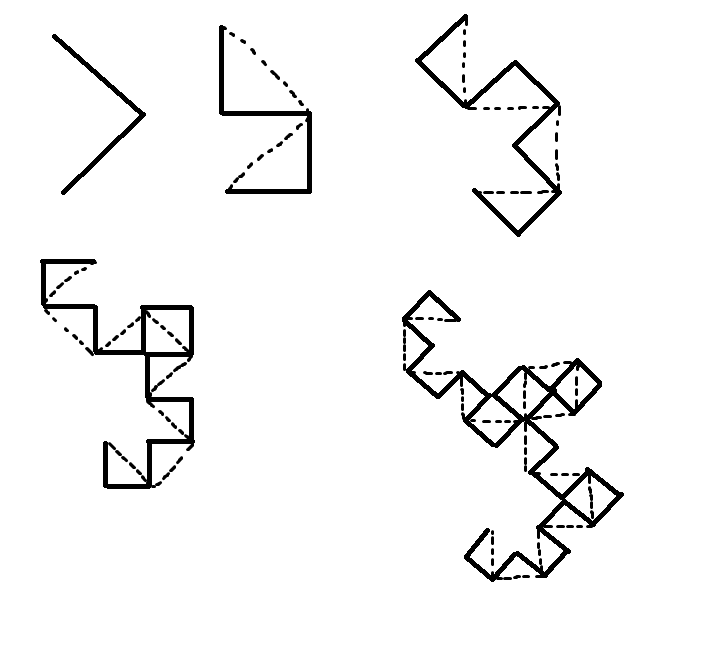


Рисунок 2 – Схема алгоритма

Код программы

Основной код.

**program** KrivoiHArter;

**uses** GraphABC;

**uses** DrawM;

**var**

i: integer;

**var**

x, y, dx, dy: integer;

**procedure** text;

**begin**

textout(10, 100, 'Left - влево');

textout(10, 120, 'Right - вправо');

textout(10, 140, 'Up - вверх');

textout(10, 160, 'Down - вниз');

textout(10, 180, 'z - увеличение');

textout(10, 200, 'x - уменьшение');

textout(10, 220, 'a - фрактал');

textout(10, 240, 'd - фрактал');

**end**;

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

**begin**

i := 0;

**case** key **of**

VK\_Up: **begin** y := y - 50 **end**;

VK\_Down: **begin** y := y + 50 **end**;

VK\_Left: **begin** x := x - 50 **end**;

VK\_Right: **begin** x := x + 50 **end**;

VK\_z:

**begin**

i := i + 1;

x := x + i;

dx := dx - i;

dy := dy + i;

y := y - i;

**end**;

Vk\_x:

**begin**

i := i - 1;

x := x - i;

dx := dx + i;

dy := dy - i;

y := y + i;

**end**;

vk\_a: **if** i > 0 **then** i -= 1;

vk\_d: **if** i < 4 **then** i += 1;

vk\_F:

**end**;

Window.Clear;

text;

draw(x, y, dx, dy);

redraw;

**end**;

**begin**

writeln('Left - влево');

writeln('Right - вправо');

writeln('Up - вверх');

writeln('Down - вниз');

writeln('z - увеличение');

writeln('x - уменьшение');

writeln('a - фрактал');

writeln('d - фрактал');

LockDrawing;

x := 200;

y := 140;

dx := 0;

dy := -4;

draw(x, y, dx, dy);

redraw;

onKeyDown += keydown;

**end**.

2 часть кода (модуль)

**unit** DrawM;

**interface**

**uses** GraphABC;

**procedure** Draw(x, y, dx, dy: integer);

**implementation**

**procedure** Draw(x, y, dx, dy: integer);

**var**

turn: **array** [1..1000] **of** Boolean;//поворот массива

a, b, d, t: integer;

f: Boolean;

i: integer;

**begin**

SetWindowSize(790, 500);//Устанавливаем ширину и высоту графического окна.

SetWindowCaption('Фракталы. Кривая Дракона');

f := true;

**for** a := 1 **to** 64 **do**

**begin**

turn[2 \* a - 1] := f;

f := **not** f;

turn[2 \* a] := turn[a];

**end**;

b := 0;

d := 1;

f := false;//при f ложном

MoveTo(x, y);

**for** a := 1 **to** 128 **do**

**begin**

**for** i := 1 **to** 127 \* 4 **do**

**begin**

b := b + d; x := x + dx; y := y + dy;

LineTo(x, y);

**if** f **and not** turn[b] **or not** f **and** turn[b] **then**

**begin**

t := dy;

dy := -dx;

**end**

**else**

**begin**

t := -dy;

dy := dx;

**end**;

dx := t;

**end**;

b := b + d; d := -d;

f := **not** f;

x := x + dx; y := y + dy;

LineTo(x, y);

**if** turn[a] **then**

**begin**

t := dy;

dy := -dx;

**end**

**else**

**begin**

t := -dy;

dy := dx;

**end**;

dx := t;

**end**;

**end**;

**end**.

Результат выполнения программы

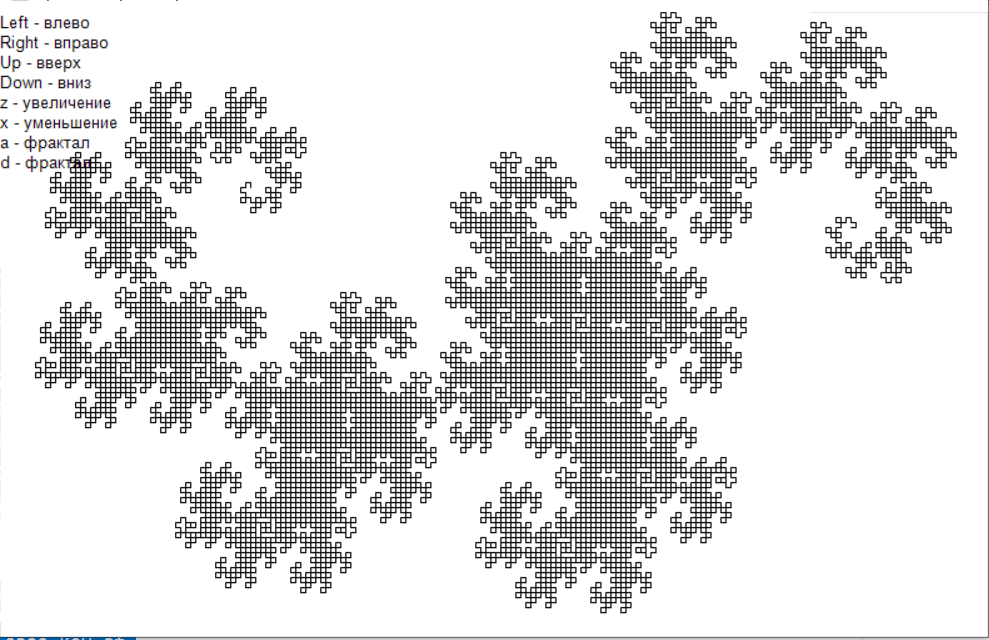


Рисунок 3 – Результат выполнения программы

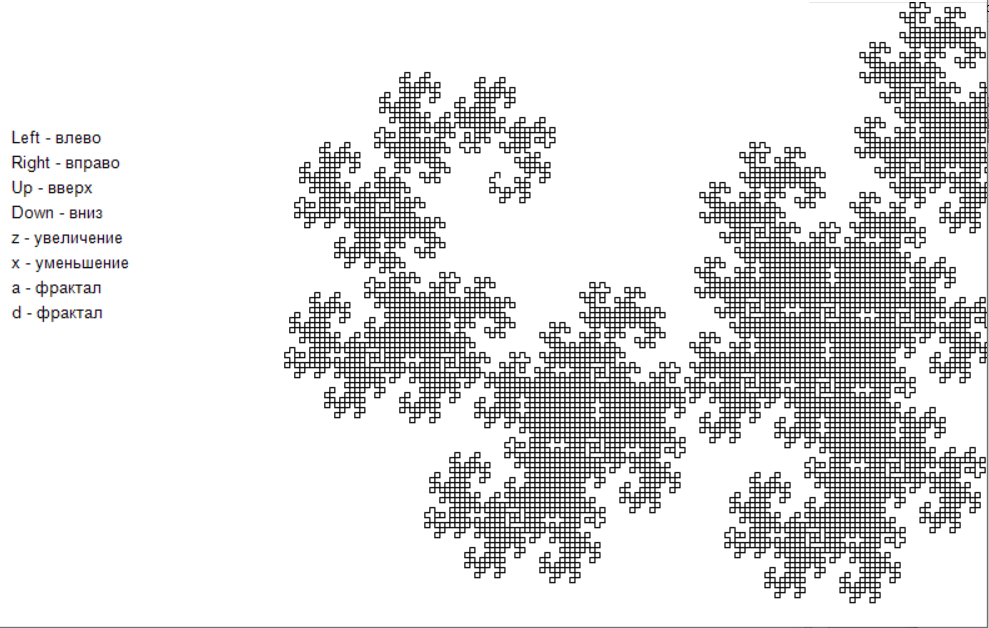


Рисунок 4 – Изменение рисунка

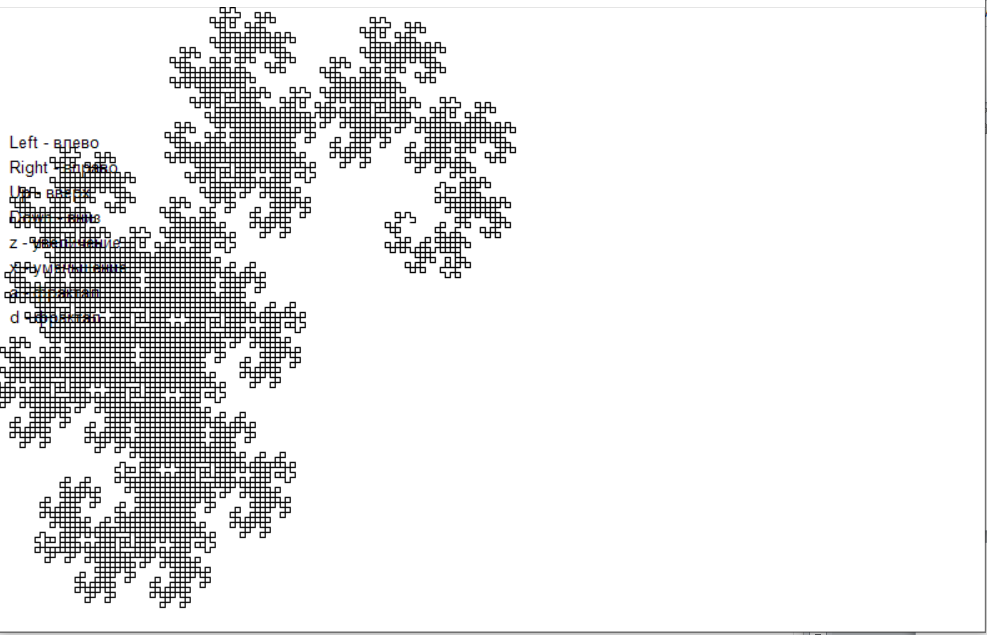


Рисунок 5 – изменение рисунка

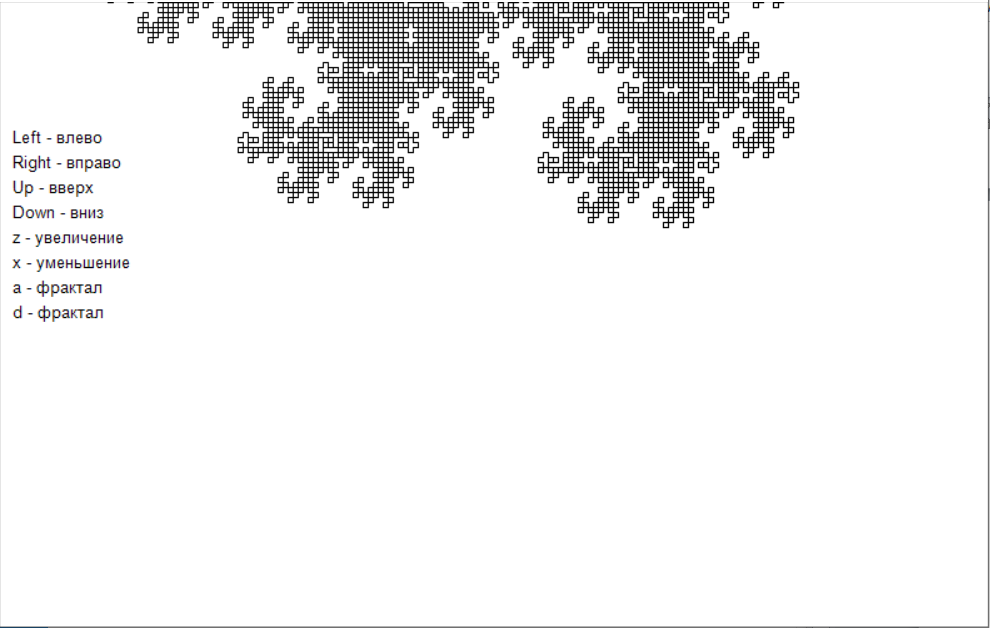


Рисунок 6 – изменение рисунка.

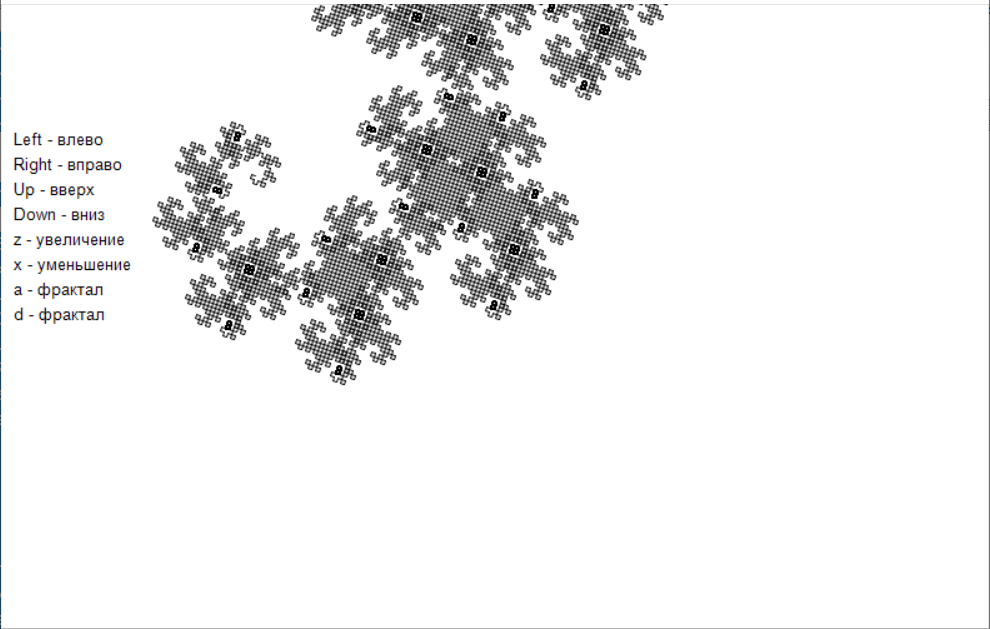


Рисунок 7 – Изменение рисунка

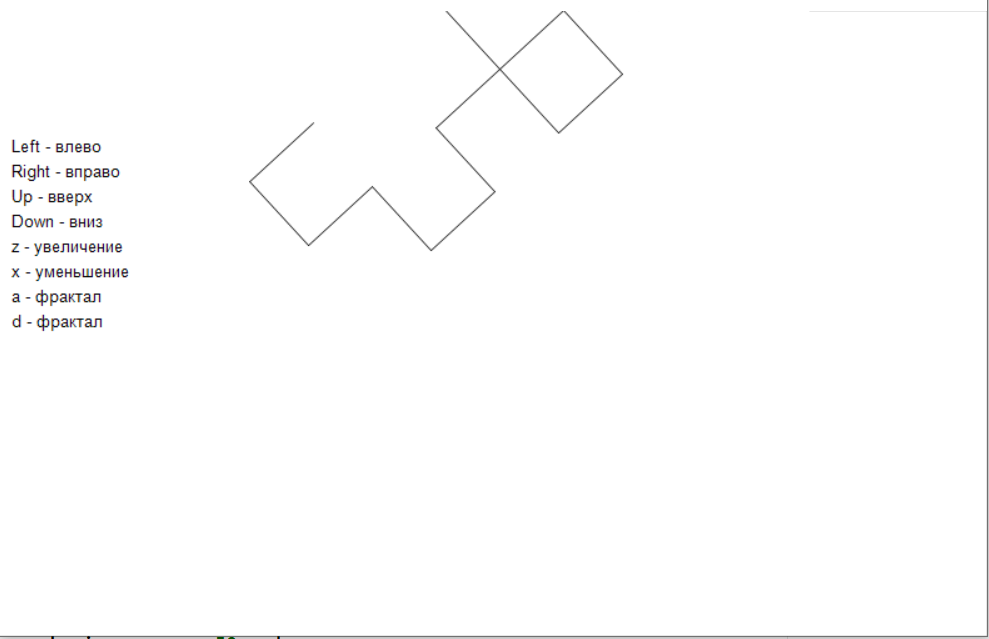


Рисунок 8 – Изменение рисунка

Вывод: В ходе выполнения работы я научился рисовать кривую Хартера-Хейтуэя. Берется отрезок, который сгибается пополам, затем эта операция повторяется многократно. После этого снова разогнуть получившуюся (сложенную) линию так, чтобы все углы были равны 90%, можно получить драконовую ломаную.

В ходе работы повторил навыки создания в графическом редакторе паскаля. Изучил новые встроенные функции в PascalABC. Научился изменять глубину прорисовки и перемещение полученной мною фигуры. Научился осуществлять полученный код в отдельном модуле. Изучил функцию Кривой Хартера-Хейтуэя.