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»**

Выполнил: студент учебной группы

ИСПк-204-52-00

Конышев Иван Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Цель работы**

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

**Формулировка задания**

* Написать программу для визуализации фрактала «Кривая Гильберта»
* Предусмотреть возможность масштабирования, изменения глубины прорисовки и перемещение полученной фигуры
* Построение множества ломанных, образующих фрактал, должно осуществляться в отдельном окне

**Описание алгоритма**

Данный алгоритм рисует фрактал «Кривая Гильберта» c реализацией возможности масштабирования, изменения глубины прорисовки.

Программа начинает с процедуры a(p), которая вызывает другие процедуры (b, c, d) в соответствии с правилами построения кривой Гильберта. Каждая из этих процедур отвечает за определенное направление движения на графике.

При нажатии определенных клавиш (стрелки и A, Z, S, X) изменяются параметры рисунка, такие как размер шага, масштаб и глубина рекурсии, что позволяет управлять выводом кривой Гильберта на экране.

Функция Redraw обеспечивает плавное перерисовывание при изменении параметров, что позволяет динамически изменять отображение фрактала на экране.

Таким образом, алгоритм создает интерактивную возможность рисовать и изменять фрактал Кривая Гильберта на экране с помощью клавиш на клавиатуре.

**Схемы алгоритма**

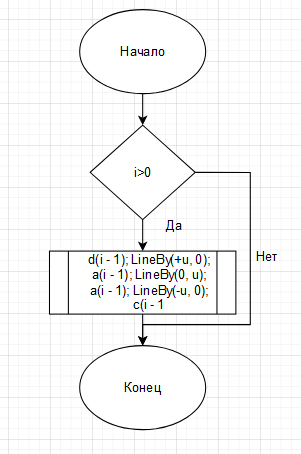


Рисунок 1 – схема алгоритма a

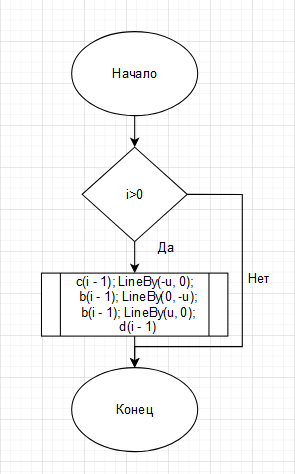


Рисунок 2 – схема алгоритма b

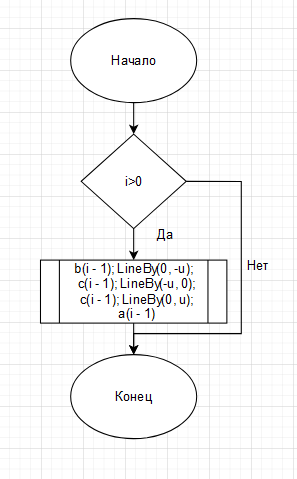


Рисунок 3 – схема алгоритма c

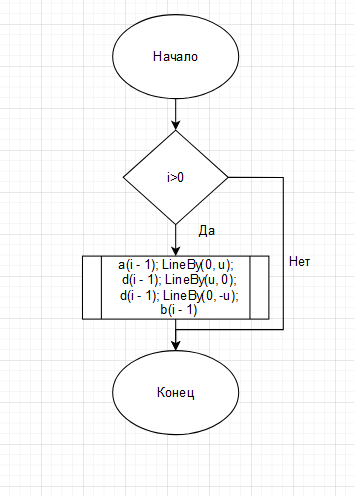


Рисунок 4 – схема алгоритма d

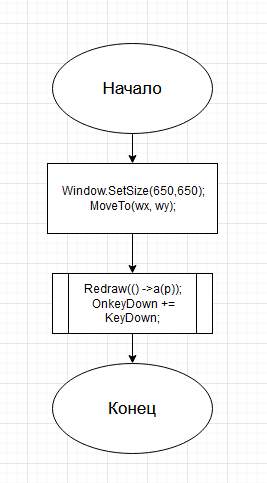


Рисунок 5 – схема алгоритма

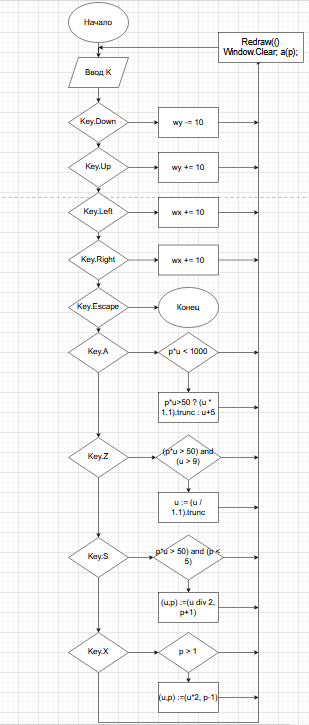


Рисунок 6 – схема алгоритма KeyDown

**Код программы**

**Uses** GraphWPF;

**var** (u,p, wx,wy) := (32, 4, 64,64);

**Procedure** a(i: Integer); **forward**;

**Procedure** b(i: Integer); **forward**;

**Procedure** c(i: Integer); **forward**;

**Procedure** d(i: Integer); **forward**;

**Procedure** a(i: Integer) :=

**If** i > 0 **Then begin**

d(i - 1); LineBy(+u, 0);

a(i - 1); LineBy(0, u);

a(i - 1); LineBy(-u, 0);

c(i - 1)

**end**;

**Procedure** b(i: integer) :=

**If** i > 0 **Then begin**

c(i - 1); LineBy(-u, 0);

b(i - 1); LineBy(0, -u);

b(i - 1); LineBy(u, 0);

d(i - 1)

**end**;

**Procedure** c(i: integer) :=

**If** i > 0 **Then begin**

b(i - 1); LineBy(0, -u);

c(i - 1); LineBy(-u, 0);

c(i - 1); LineBy(0, u);

a(i - 1)

**end**;

**Procedure** d(i: integer) :=

**If** i > 0 **Then begin**

a(i - 1); LineBy(0, u);

d(i - 1); LineBy(u, 0);

d(i - 1); LineBy(0, -u);

b(i - 1)

**end**;

**Procedure** KeyDown(k: Key);

**begin**

**case** K **of**

Key.Down: wy -= 10;

Key.Up: wy += 10;

Key.Left: wx += 10;

Key.Right: wx -= 10;

Key.Escape: halt(1); //выход

Key.A: **if** p\*u < 1000 **then** u := p\*u>50 ? (u \* 1.1).trunc : u+5;

Key.Z: **if** (p\*u > 50) **and** (u > 9) **then** u := (u / 1.1).trunc;

Key.S: **if** (p\*u > 50) **and** (p < 5) **then** (u,p) :=(u **div** 2, p+1);

Key.X: **if** p > 1 **then** (u,p) :=(u\*2, p-1);

**end**;

MoveTo(wx,wy);

Redraw(() ->**begin** Window.Clear; a(p); **end**);

**end**;

**begin**

Window.SetSize(650,650);

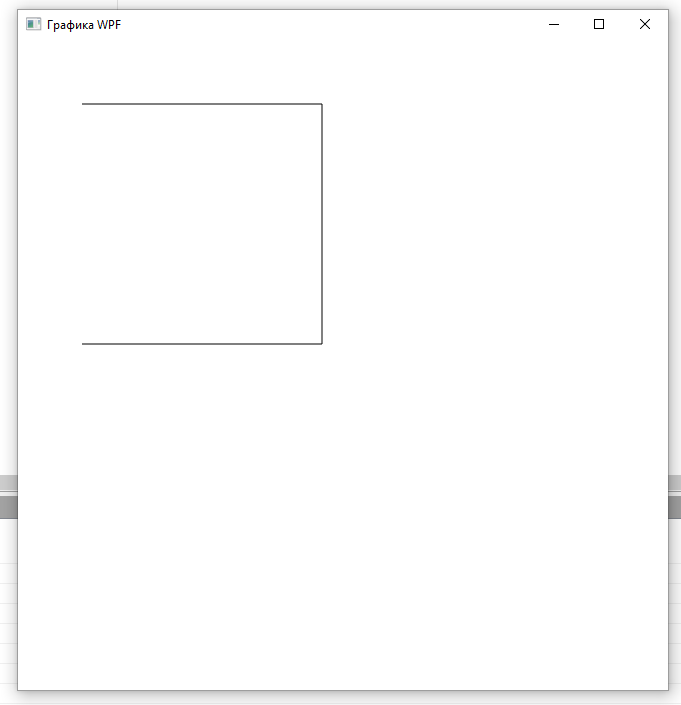
MoveTo(wx, wy);

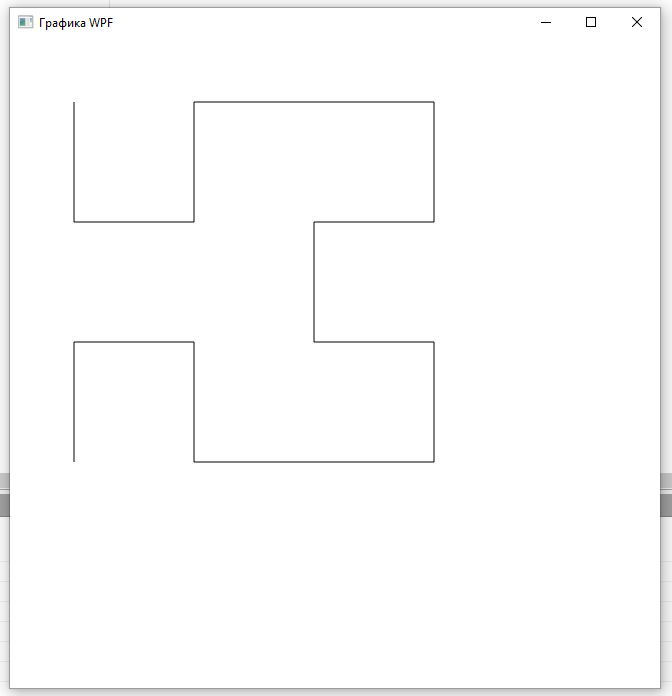
Redraw(() ->a(p));

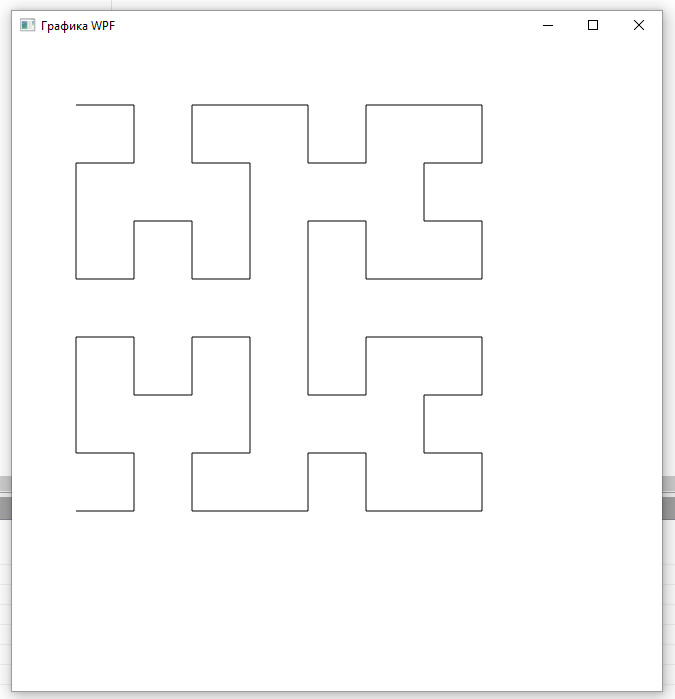
OnkeyDown += KeyDown;

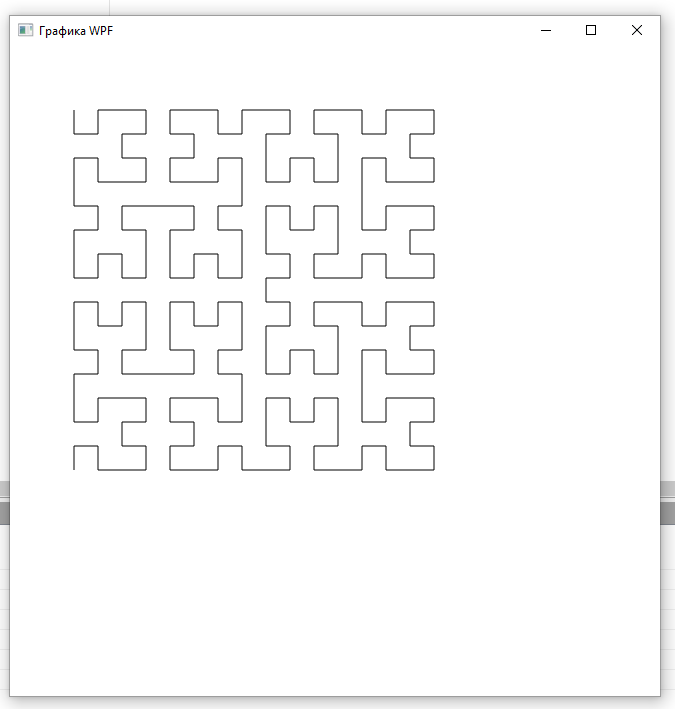
**end**.

**Результат выполнения программы**









**Вывод**

В ходе выполнения работы были приобретены следующие навыки:

* Реализация рекурсивных алгоритмов на языке программирования.
* Понимание принципов работы рекурсивных алгоритмов.
* Использование рекурсии для решения различных задач.

Кроме того, было получено представление о фракталах, их свойствах и способах построения.

Таким образом, полученные навыки и знания могут быть использованы для дальнейшего изучения и разработки алгоритмов, а также для исследования фракталов.