Министерство науки и высшего образования Россйиской Федеражии

Фкдкральное гоусдарственное бюджетное образовательное учереждение высшего образования

Вятский государственный университет

Коллкдж ВятГУ

ОТЧЕТ

ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №1

«ИССЛЕДОВАНИЕ ФРАКТАЛОВ»

ПО

«МДК 05.02 Разработка кода информационных систем»

Выполнил: студент учебной группы

ИСПк-202-52-00

Шипицын Евгений Алексеевич

Преподаватель:

Сергеева Елизавета Григорьевна

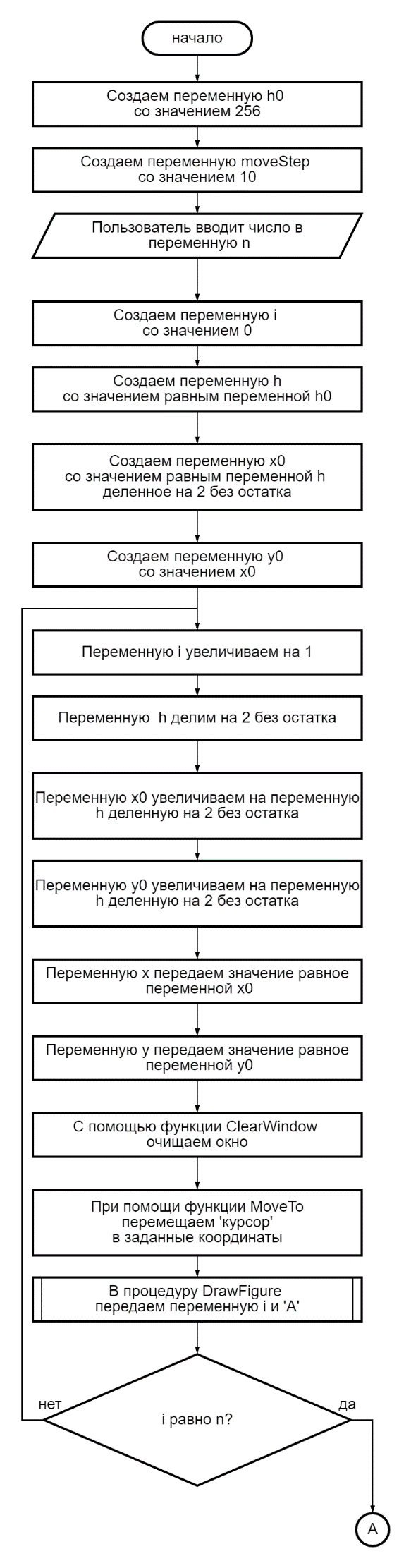
Цель работы: получение навыков реализации алгоритмов с рекурсивными вычислениями.

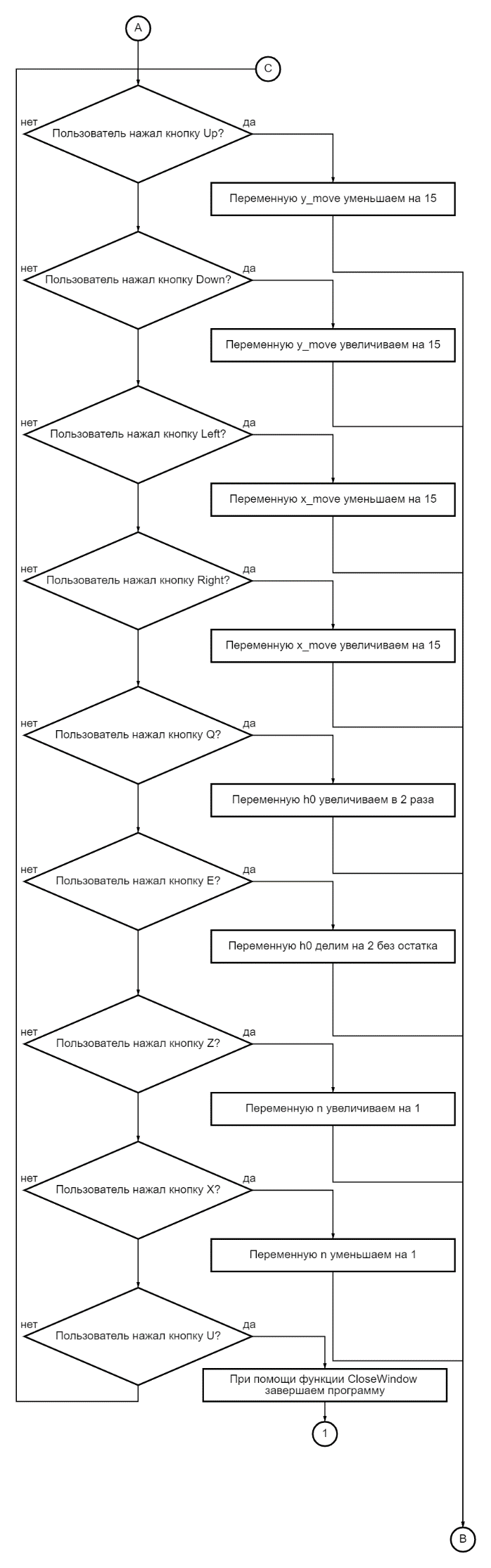
Задание:

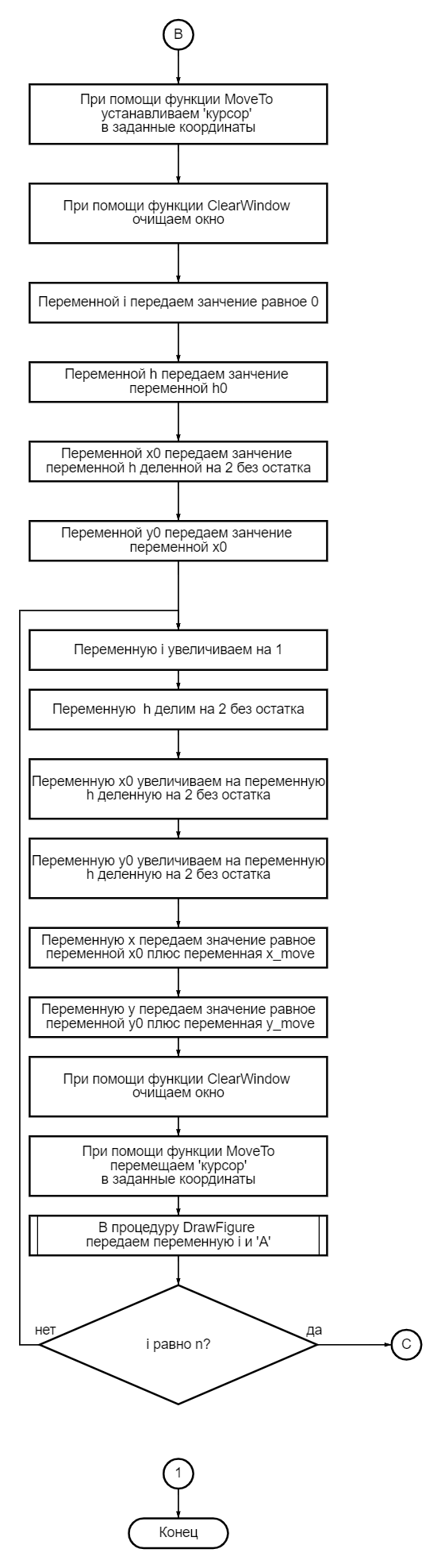
1. Написать программу для визуализации фрактала “Кривая Гильберта”
2. Предусмотреть возможность масштабирования, изменения глубины прорисовки и перемещения полученной фигуры.
3. Построение множества ломанных, образующих фрактал, должно осуществляться в отдельном модуде.

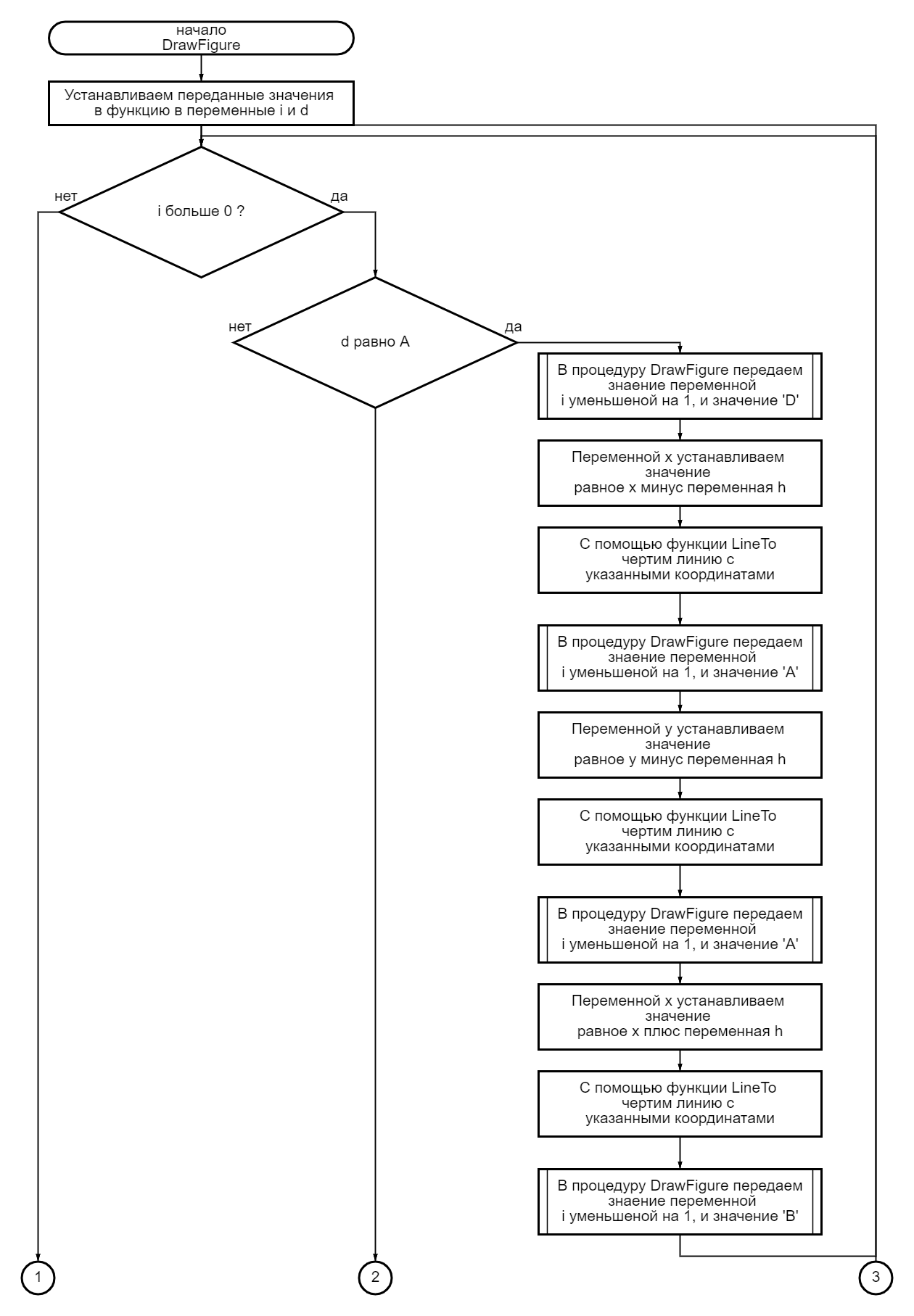
Программа:

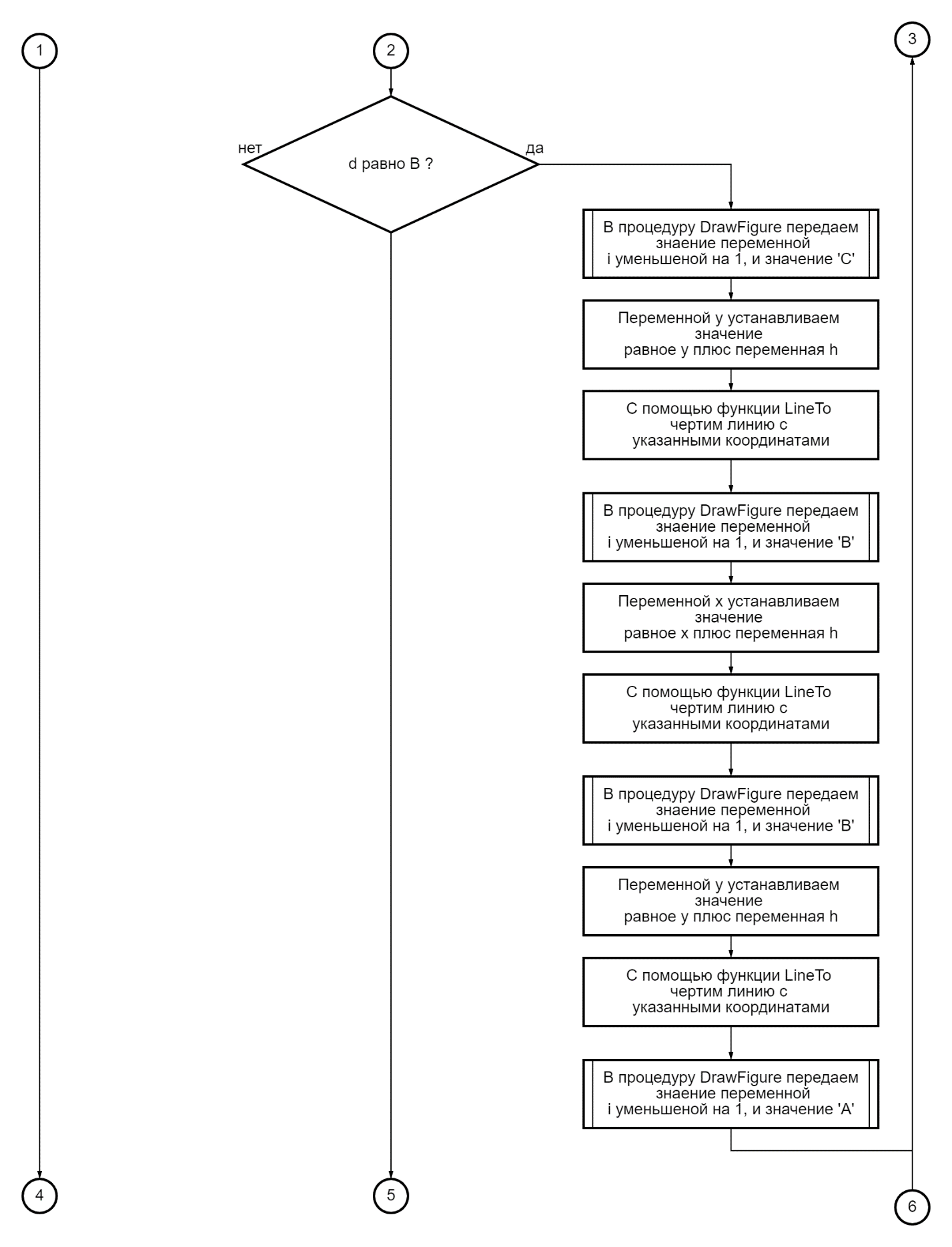
1. Пользователь вводит значение для переменной n, которая указывает глубину рекурсии.
2. Программа рисует фрактал с глубиной (детализация фигуры), которую указал пользователь.
3. Пока пользователь не нажал кнопку U :
   1. Если пользователь нажал кнопку Up, то происходит перемещение фигуры вверх;
   2. Если пользователь нажал кнопку Down, то происходит перемещение фигуры вниз;
   3. Если пользователь нажал кнопку Left, то происходит перемещение фигуры влево;
   4. Если пользователь нажал кнопку Right, то происходит перемещение фигуры вправо;
   5. Если пользователь нажал кнопку X, то происходит уменьшение глубины рекурсии;
   6. Если пользователь нажал кнопку Z, то происходит увеличение глубины рекурсии;
   7. Если пользователь нажал кнопку E, то происходит приближение фигур;
   8. Если пользователь нажал кнопку Q, то происходит приближение фигуры;

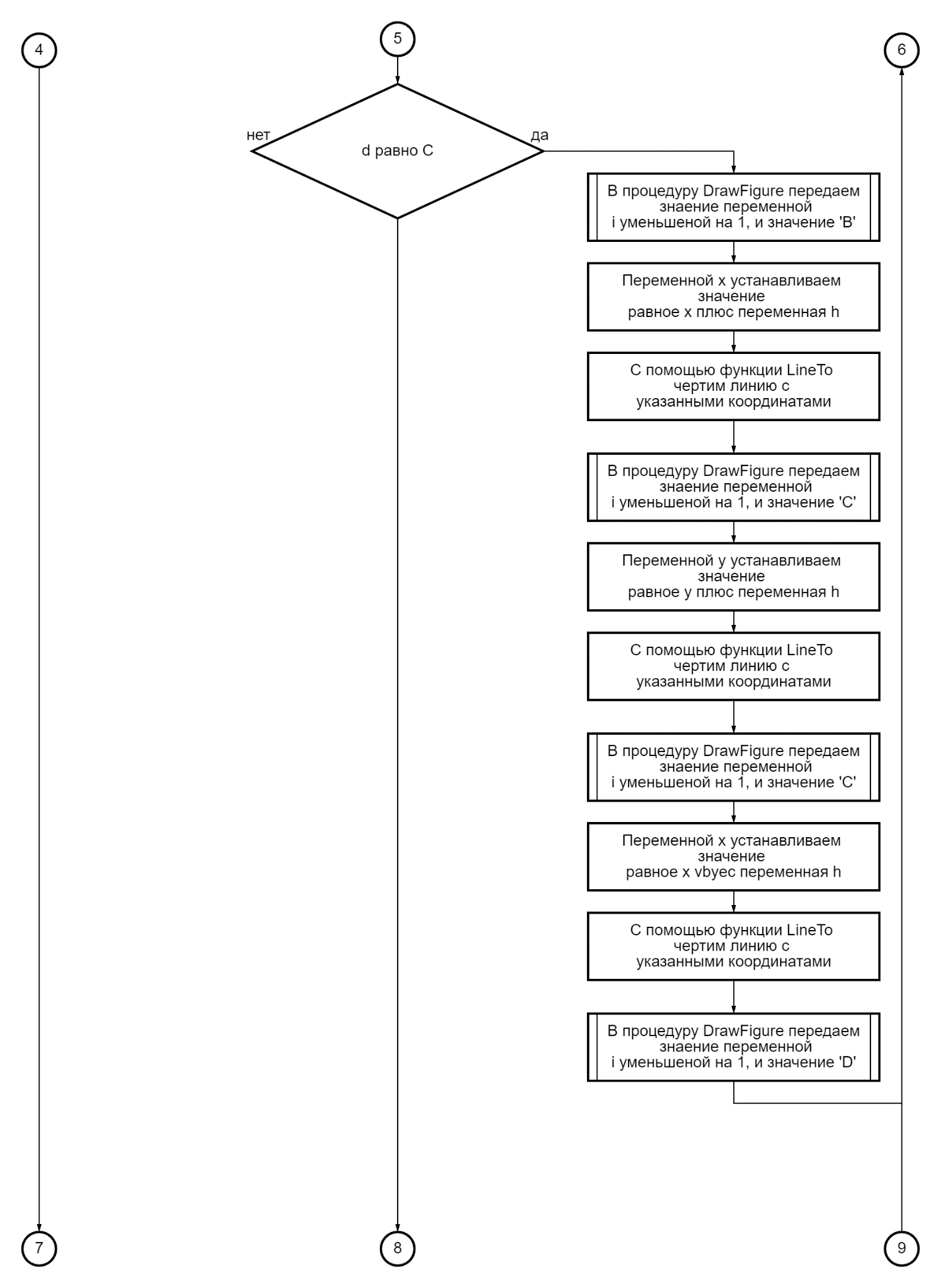


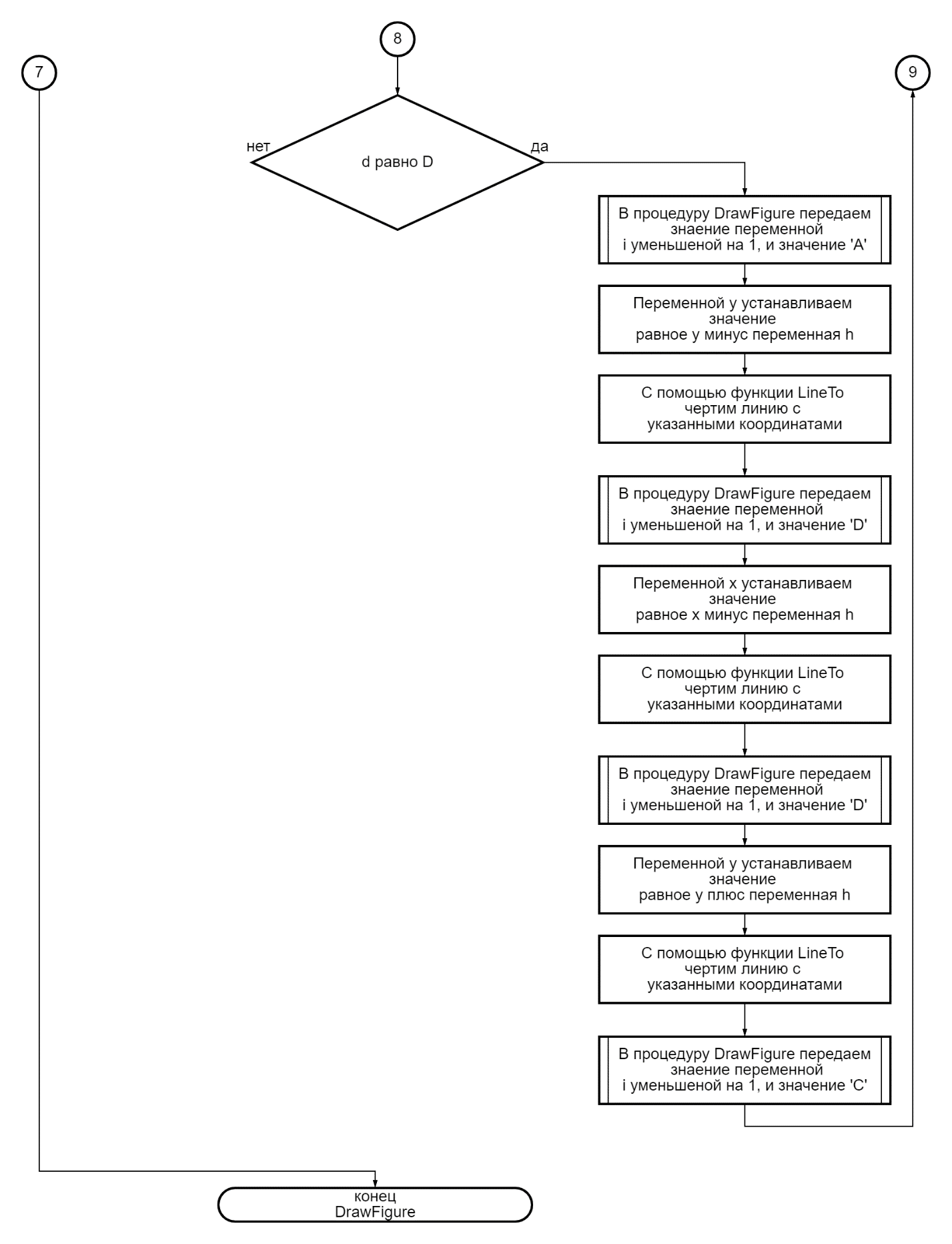












Код программы:

**Program** rek;

**uses** graphABC;

**var**

i, h, x, y, y\_move, x\_move, x0, y0, n, h0, scaleFactor, moveStep: integer;

**Procedure** DrawFigure(i: integer; d: char);

**begin**

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

**begin**

**case** d **of**

'A':

**begin**

DrawFigure(i - 1, 'D');

x := x - h;

LineTo(x, y);

DrawFigure(i - 1, 'A');

y := y - h;

LineTo(x, y);

DrawFigure(i - 1, 'A');

x := x + h;

LineTo(x, y);

DrawFigure(i - 1, 'B');

**end**;

'B':

**begin**

DrawFigure(i - 1, 'C');

y := y + h; LineTo(x, y);

DrawFigure(i - 1, 'B');

x := x + h; LineTo(x, y);

DrawFigure(i - 1, 'B');

y := y - h; LineTo(x, y);

DrawFigure(i - 1, 'A');

**end**;

'C':

**begin**

DrawFigure(i - 1, 'B');

x := x + h; LineTo(x, y);

DrawFigure(i - 1, 'C');

y := y + h; LineTo(x, y);

DrawFigure(i - 1, 'C');

x := x - h; LineTo(x, y);

DrawFigure(i - 1, 'D');

**end**;

'D':

**begin**

DrawFigure(i - 1, 'A');

y := y - h; LineTo(x, y);

DrawFigure(i - 1, 'D');

x := x - h; LineTo(x, y);

DrawFigure(i - 1, 'D');

y := y + h; LineTo(x, y);

DrawFigure(i - 1, 'C');

**end**;

**end**;

**end**;

**end**;

**begin**

h0 := 256;

moveStep := 10;

writeln('');

readln(n);

i := 0;

h := h0;

x0 := h **div** 2;

y0 := x0;

**repeat**

i := i + 1;

h := h **div** 2;

x0 := x0 + (h **div** 2);

y0 := y0 + (h **div** 2);

x := x0;

y := y0;

MoveTo(x, y);

DrawFigure(i, 'A');

**until** i = n;

OnKeyDown := **procedure**(key: integer) ->

**begin**

**case** key **of**

VK\_UP: y\_move := y\_move - 15;

VK\_DOWN: y\_move := y\_move + 15;

VK\_LEFT: x\_move := x\_move - 15;

VK\_RIGHT: x\_move := x\_move + 15;

VK\_Q: h0 := h0 \* 2;

VK\_E: h0 := h0 **div** 2;

VK\_Z: n := n + 1;

VK\_X: n := n - 1;

VK\_U: CloseWindow;

**end**;

MoveTo(x, y);

ClearWindow;

i := 0;

h := h0;

x0 := h **div** 2;

y0 := x0;

**repeat**

i := i + 1;

h := h **div** 2;

x0 := x0 + (h **div** 2);

y0 := y0 + (h **div** 2);

x := x0 + x\_move;

y := y0 + y\_move;

MoveTo(x, y);

DrawFigure(i, 'A');

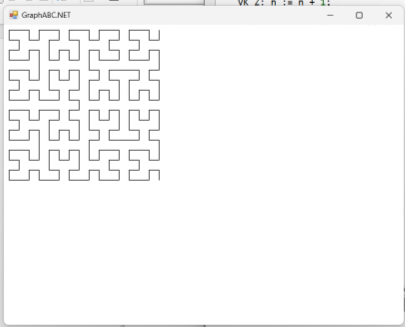
**until** i = n;

**end**;

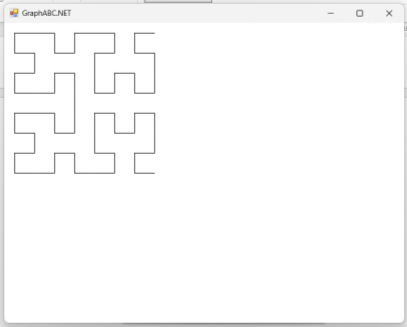
readln;

**end**.

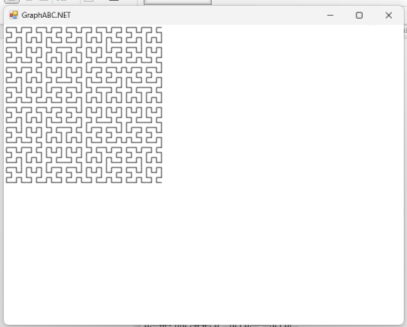
Программа вывод следующее, если ввести n равное 4:



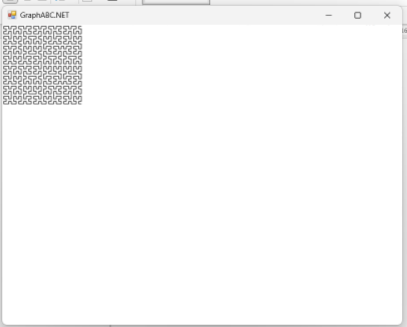
Программа вывод следующее, если нажать кнопку X:



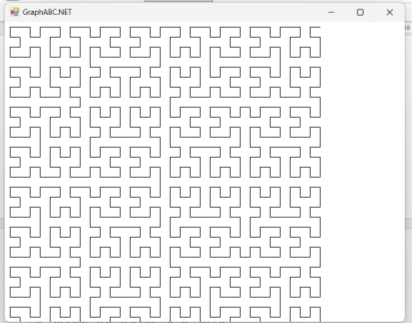
Программа вывод следующее, если нажать кнопку Z 2 раза:



Программа вывод следующее, если нажать кнопку E:



Программа вывод следующее, если нажать кнопку Q два раза:



Вывод:

Работа с рекурсией при создании здания по кривой Гильберта является увлекательным и интересным занятием. Рекурсивные алгоритмы позволяют создавать сложные и красивые структуры, отражающие математическую красоту кривой Гильберта. Этот процесс также способствует развитию навыков программирования и понимания алгоритмов.

Но в тоже время возникли некоторые сложности, а именно понять алгоритм построения узоров фрактала, в зависимости от глубины, при помощи рекурсии.