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 ФАКТОРИАЛА»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-101-51-00

Орлов Валерий Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. Цель работы:

Получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами

1. Скриншот задания (с вариантом)

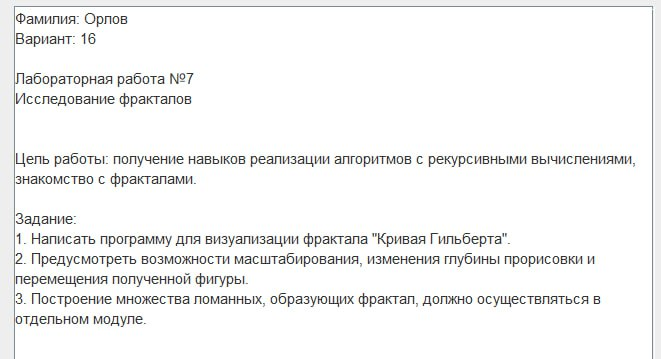


Рисунок 1 – Скриншот задания

1. Описание алгоритма

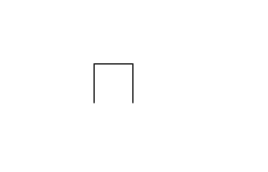


Рисунок 2 – Первый этап

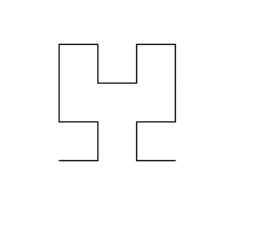


Рисунок 3 – Второй этап

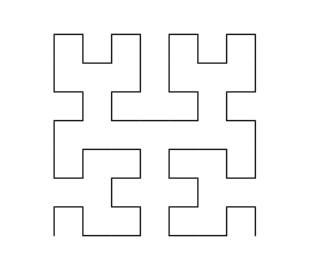


Рисунок 4 – Третий этап

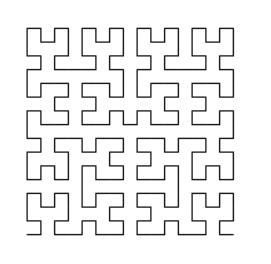


Рисунок 5 – Четвертый этап

"Кривая Гильберта" строится следующим образом: начинается она с одного отрезка. На первой итерации этот отрезок преобразуется в зигзагообразную ломаную, напоминающую букву "U". На каждом следующем шаге каждый отрезок этой ломаной заменяется уменьшенной копией всей фигуры предыдущего порядка, повернутой и отраженной соответствующим образом. Так, после второй итерации каждый из изначальных отрезков заменяется фигурой, состоящей из 4 меньших "U"-образных звеньев, соединенных между собой. И так далее: на каждом шаге каждый отрезок ломаной предыдущего порядка заменяется фигурой, которая уже состоит из 16 более мелких отрезков. Процесс повторяется рекурсивно, и с каждой итерацией кривая Гильберта заполняет пространство всё более плотно, приближаясь к пределу, который занимает непрерывную площадь, но при этом имеет нулевую площадь.

1. Код программы
2. **uses** GraphABC, Fractal\_Module;
4. **var**
5. x0, y0, x00, y00, mas, n: integer;
6. **function** KU(i:integer):integer; **forward**;
7. **function** KD(i:integer):integer; **forward**;
8. **function** KR(i:integer):integer; **forward**;
9. **function** KL(i:integer):integer; **forward**;
10. **function** per: integer; **forward**;
12. **function** KU(i: integer): integer;
13. **begin**
14. **if** i > 0 **then**
15. **begin**
16. KR(i - 1);
17. LineUP(x0, y0, mas);
18. KU(i - 1);
19. LineRIGHT(x0, y0, mas);
20. KU(i - 1);
21. LineDOWN(x0, y0, mas);
22. KL(i - 1);
23. **end**;
24. **end**;
26. **function** KD(i: integer): integer;
27. **begin**
28. **if** i > 0 **then**
29. **begin**
30. KL(i - 1);
31. LineDOWN(x0, y0, mas);
32. KD(i - 1);
33. LineLEFT(x0, y0, mas);
34. KD(i - 1);
35. LineUP(x0, y0, mas);
36. KR(i - 1);
37. **end**;
38. **end**;
40. **function** KL(i: integer): integer;
41. **begin**
42. **if** i > 0 **then**
43. **begin**
44. KD(i - 1);
45. LineLEFT(x0, y0, mas);
46. KL(i - 1);
47. LineDOWN(x0, y0, mas);
48. KL(i - 1);
49. LineRIGHT(x0, y0, mas);
50. KU(i - 1);
51. **end**;
52. **end**;
54. **function** KR(i: integer): integer;
55. **begin**
56. **if** i > 0 **then**
57. **begin**
58. KU(i - 1);
59. LineRIGHT(x0, y0, mas);
60. KR(i - 1);
61. LineUP(x0, y0, mas);
62. KR(i - 1);
63. LineLEFT(x0, y0, mas);
64. KD(i - 1);
65. **end**;
66. **end**;
68. **procedure** Key(key: integer);
69. **begin**
70. **case** key **of**
71. VK\_Up : **begin** y00 := y00 - 100; per; **end**;
72. VK\_Down : **begin** y00 := y00 + 100; per; **end**;
73. VK\_Left : **begin** x00 := x00 - 100; per; **end**;
74. VK\_Right : **begin** x00 := x00 + 100; per; **end**;
75. $41 : **begin** mas := mas + 2; per; **end**; // 'A' - увеличить масштаб
76. $5A : **begin if** mas > 0 **then** mas := mas - 2; per; **end**; // 'Z' - уменьшить масштаб
77. $53 : **begin if** n < 6 **then** n := n + 1; per; **end**; // 'S' - увеличить фрактал
78. $58 : **begin if** n > 0 **then** n := n - 1; per; **end**; // 'X' - уменьшить фрактал
79. **end**;
80. **end**;
82. **function** per: integer;
83. **begin**
84. LockDrawing;
85. ClearWindow;
86. x0 := x00;
87. y0 := y00;
88. KU(n);
89. Redraw;
90. **end**;
92. **begin**
93. n := 1;
94. x00 := 270;
95. y00 := 220;
96. mas := 15;
97. per;
98. OnKeyDown := Key;
99. end.

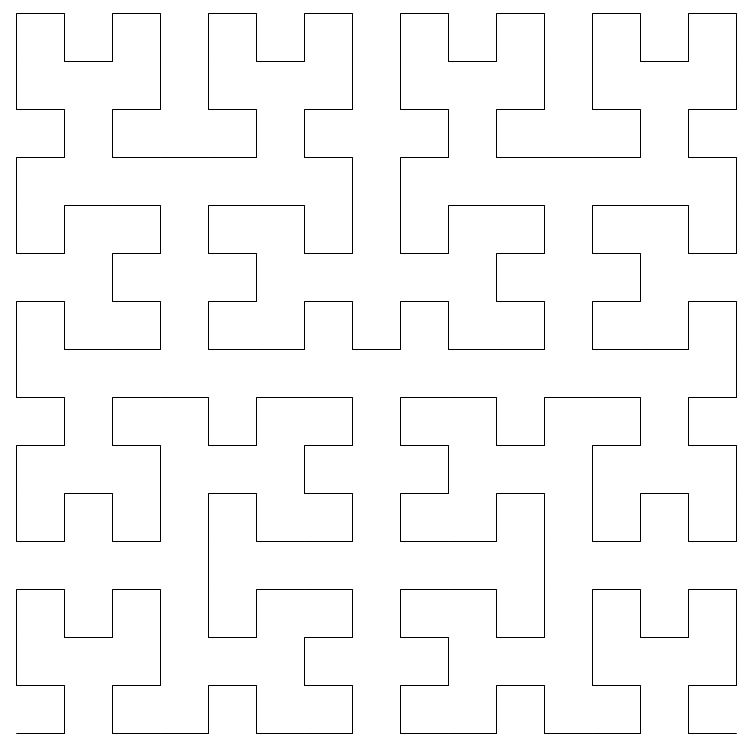


Рисунок 6 – Результат выполнения программы

1. Вывод

В данном отчете представлены данные и решения, которые были использованы в домашней контрольной работе по программированию на языке Pascal. Задача включала создание фрактала, его масштабирование и перемещение.

.