Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №1**

**«Исследование фрактала»**

**МДК 05.02 Разработка кода информационных систем**

Выполнил: студент учебной группы

ИСПк-203-52-00

Алферова Софья

Преподаватель:

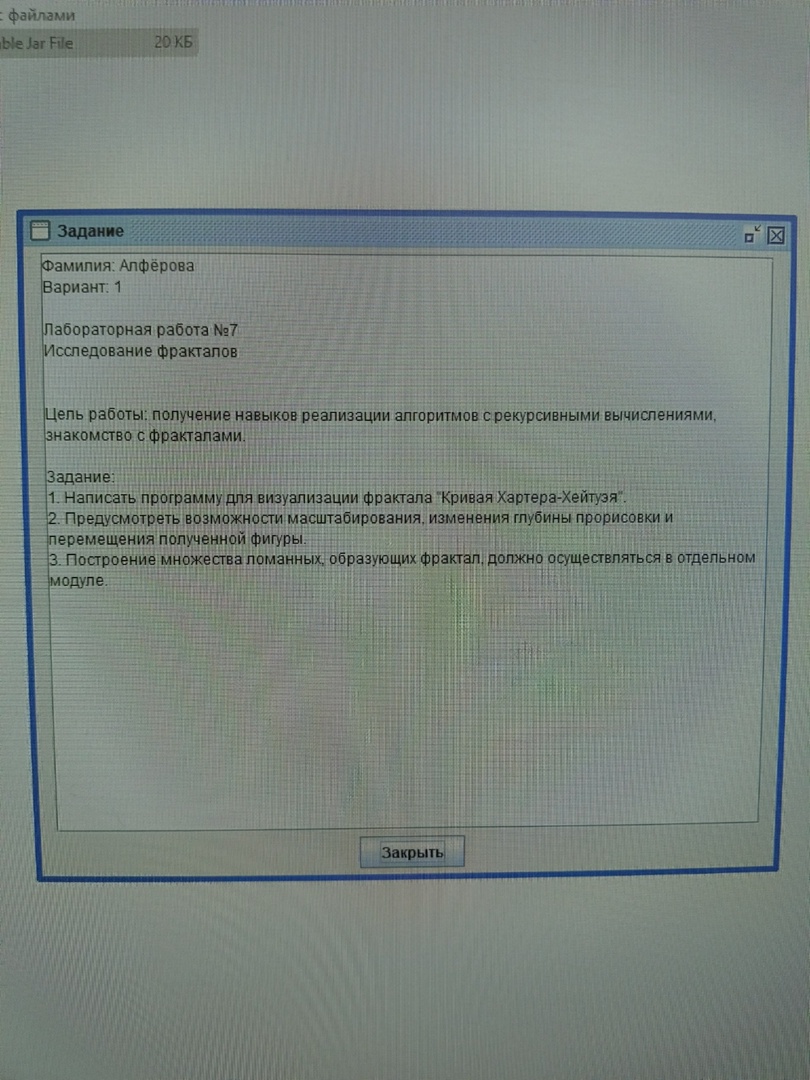
Сергеева Елизавета Григорьевна

Киров

2024

В отчете должны отображаться:

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



1. Описание алгоритма

* В процедуре Draw(x, y, dx, dy) реализуется сам процесс рисования фрактала. В ней устанавливаются размеры окна и заголовок окна.
* Заполнение массива turn, задаются начальные значения переменных b и d, а также переменная f. Далее происходит окраска пикселей за счет последовательного соединения линий.
* В процедуре text выводятся инструкции для управления фракталом на экране.
* Процедура KeyDown отвечает за реакцию на нажатие клавиш. В зависимости от нажатой клавиши происходят изменения координат точки начала рисования фрактала, а также увеличение или уменьшение масштаба фрактала.
* В основной части программы задаются начальные значения переменных x, y, dx, dy и вызывается процедура рисования фрактала.

1. Схема алгоритма с комментариями

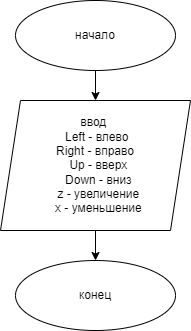


Рис1. «Схема алгоритма подпрограммы DrawText»

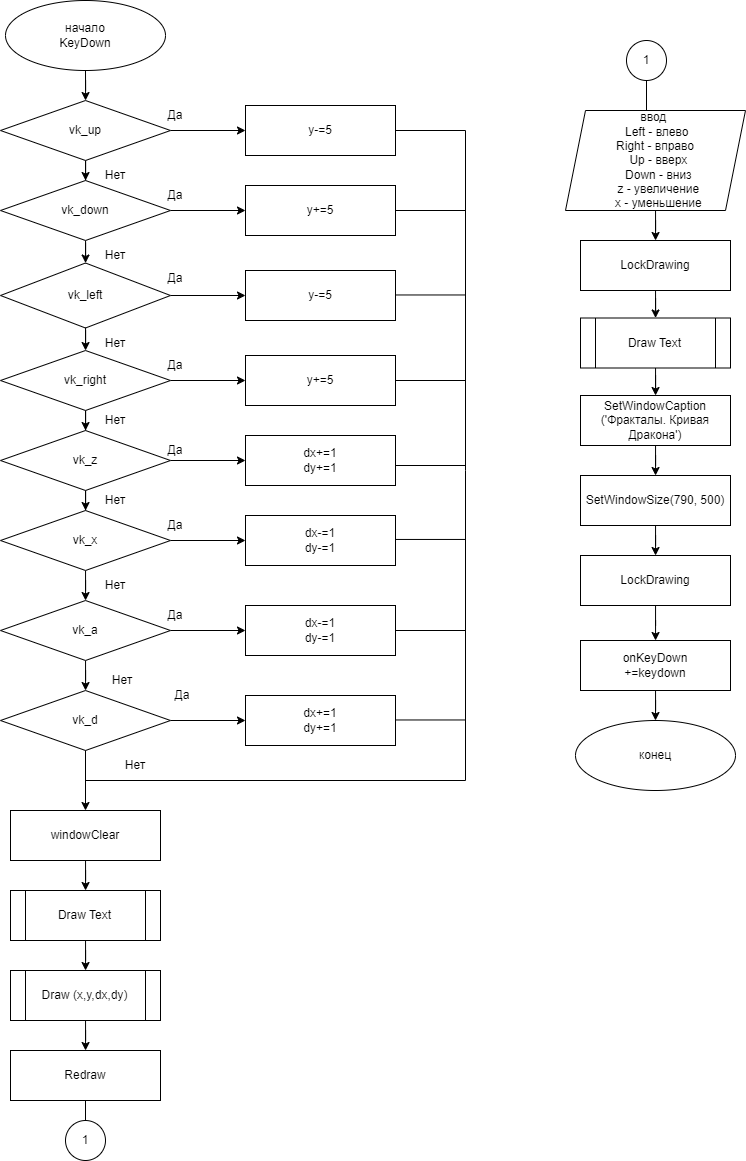


Рис. 2 «Схема алгоритма подпрограммы KeyDown»

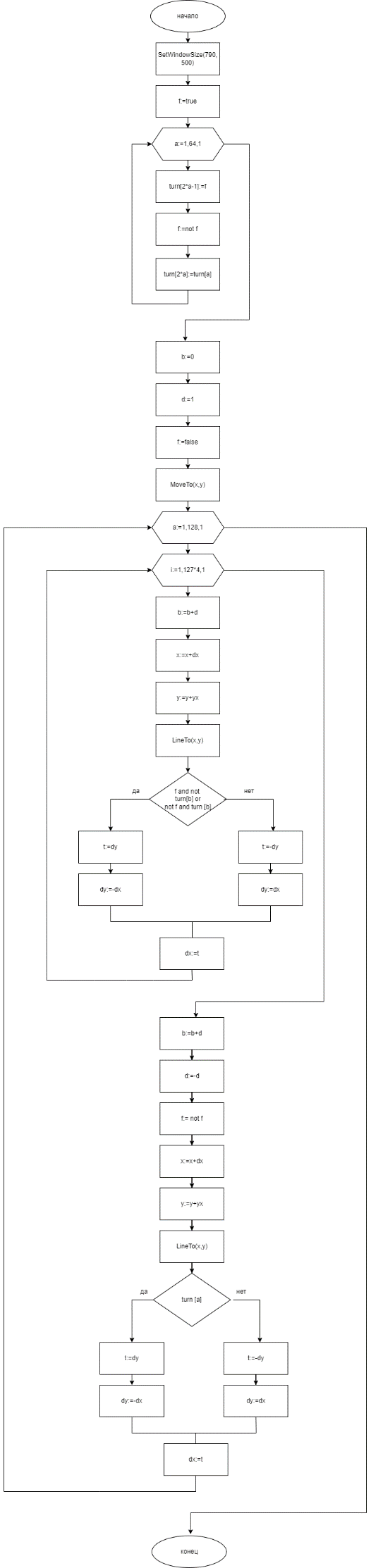
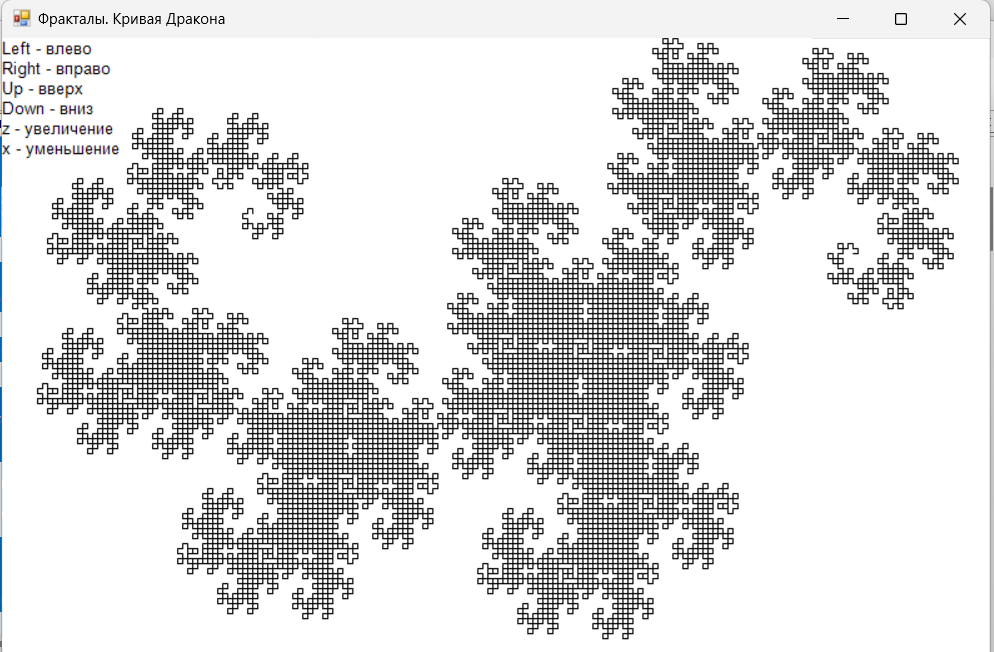


Рис. 3 «Схема алгоритма модуля»

1. Код программы
2. **program** fractal;
3. **uses** GraphABC;
4. **var**
5. i: integer;
6. **procedure** Draw(x, y, dx, dy: integer);
7. **var**
8. turn: **array** [1..1000] **of** Boolean; // поворот массива
9. a, b, d, t: integer;
10. f: Boolean;
11. i: integer;
12. **begin**
13. SetWindowSize(790, 500); // Устанавливаем ширину и высоту графического окна.
14. SetWindowCaption('Фракталы. Кривая Дракона');
15. f := true;
16. **for** a := 1 **to** 64 **do**
17. **begin**
18. turn[2 \* a - 1] := f;
19. f := **not** f;
20. turn[2 \* a] := turn[a];
21. **end**;
22. b := 0;
23. d := 1;
24. f := false; // при f ложном
25. MoveTo(x, y);
26. **for** a := 1 **to** 128 **do**
27. **begin**
28. **for** i := 1 **to** 127 \* 4 **do**
29. **begin**
30. b := b + d; x := x + dx; y := y + dy;
31. LineTo(x, y);
32. **if** f **and not** turn[b] **or not** f **and** turn[b] **then**
33. **begin**
34. t := dy;
35. dy := -dx;
36. **end**
37. **else**
38. **begin**
39. t := -dy;
40. dy := dx;
41. **end**;
42. dx := t;
43. **end**;
44. b := b + d; d := -d;
45. f := **not** f;
46. x := x + dx; y := y + dy;
47. LineTo(x, y);
48. **if** turn[a] **then**
49. **begin**
50. t := dy;
51. dy := -dx;
52. **end**
53. **else**
54. **begin**
55. t := -dy;
56. dy := dx;
57. **end**;
58. dx := t;
59. **end**;
60. **end**;
61. **var**
62. x, y, dx, dy, k: integer;
63. **procedure** text;
64. **begin**
65. textout(10, 100, 'Left - влево');
66. textout(10, 120, 'Right - вправо');
67. textout(10, 140, 'Up - вверх');
68. textout(10, 160, 'Down - вниз');
69. textout(10, 180, 'z - увеличение');
70. textout(10, 200, 'x - уменьшение');
71. **end**;
72. **procedure** KeyDown(key: integer);
73. **begin**
74. **case** key **of**
75. VK\_Up: **begin** y := y - 5; **end**;
76. VK\_Down: **begin** y := y + 5; **end**;
77. VK\_Left: **begin** x := x - 5; **end**;
78. VK\_Right: **begin** x := x + 5; **end**;
79. VK\_z: **begin** dx := dx + 1; dy := dy + 1; **end**; // увеличение
80. Vk\_x: **begin** dx := dx - 1; dy := dy - 1; **end**; // уменьшение
81. VK\_a: **begin** dx := dx - 1; dy := dy - 1; **end**; // движение назад
82. VK\_d: **begin** dx := dx + 1; dy := dy + 1; **end**; // движение вперед
83. **end**;
84. Window.Clear;
85. text;
86. draw(x, y, dx, dy);
87. redraw;
88. **end**;
89. **begin**
90. writeln('Left - влево');
91. writeln('Right - вправо');
92. writeln('Up - вверх');
93. writeln('Down - вниз');
94. writeln('z - увеличение');
95. writeln('x - уменьшение');
96. LockDrawing;
97. x := 200;
98. y := 140;
99. dx := 0;
100. dy := -4;
101. draw(x, y, dx, dy);
102. redraw;
103. onKeyDown += keydown;
104. **end**.

6. Результат выполнения программы



7. Вывод

Таким образом, мы получили навыки реализации алгоритмов с рекурсивными вычислениями. Также узнали, что такое фрактал и как самостоятельно его реализовать.