Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №4**

**«РАБОТА В ГРАФИЧЕСКОМ РЕЖИМЕ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Гущин Тимофей Михайлович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель работы:**

Освоить принципы работы в графическом режиме; получить базовые навыки взаимодействия с графическими примитивами

1. **Формулировка задания:**

Вариант: 8

1. Дополнить программу, реализованную в ходе предыдущей лабораторной работы, режимом визуализации.
2. ﻿﻿﻿Предусмотреть возможность вывода кривой, ограничивающей фигуру, на координатную плоскость.
3. ﻿﻿﻿Реализовать следующие возможности и элементы: масштабирование графика, подписи на осях, вывод информации о задании
4. Реализовать не менее двух возможностей из представленных независимое масштабирование по осям, штриховка вычисляемой площади, визуализация численного расчета интеграла.
5. **Описание алгоритма:**

Данный алгоритм состоит из двух процедур. Процедура inf предназначена для выведения информации в модуль clrt. Вторая процедура graph, предназначена для графического отображения графика и выделения площади криволинейной трапеции. Так же задействована процедура left\_method, которая была выполнена ранее. В итоге создано кейс-меню, из которого можно управлять всеми действиями в программе.

1. **Схемы алгоритмов:**

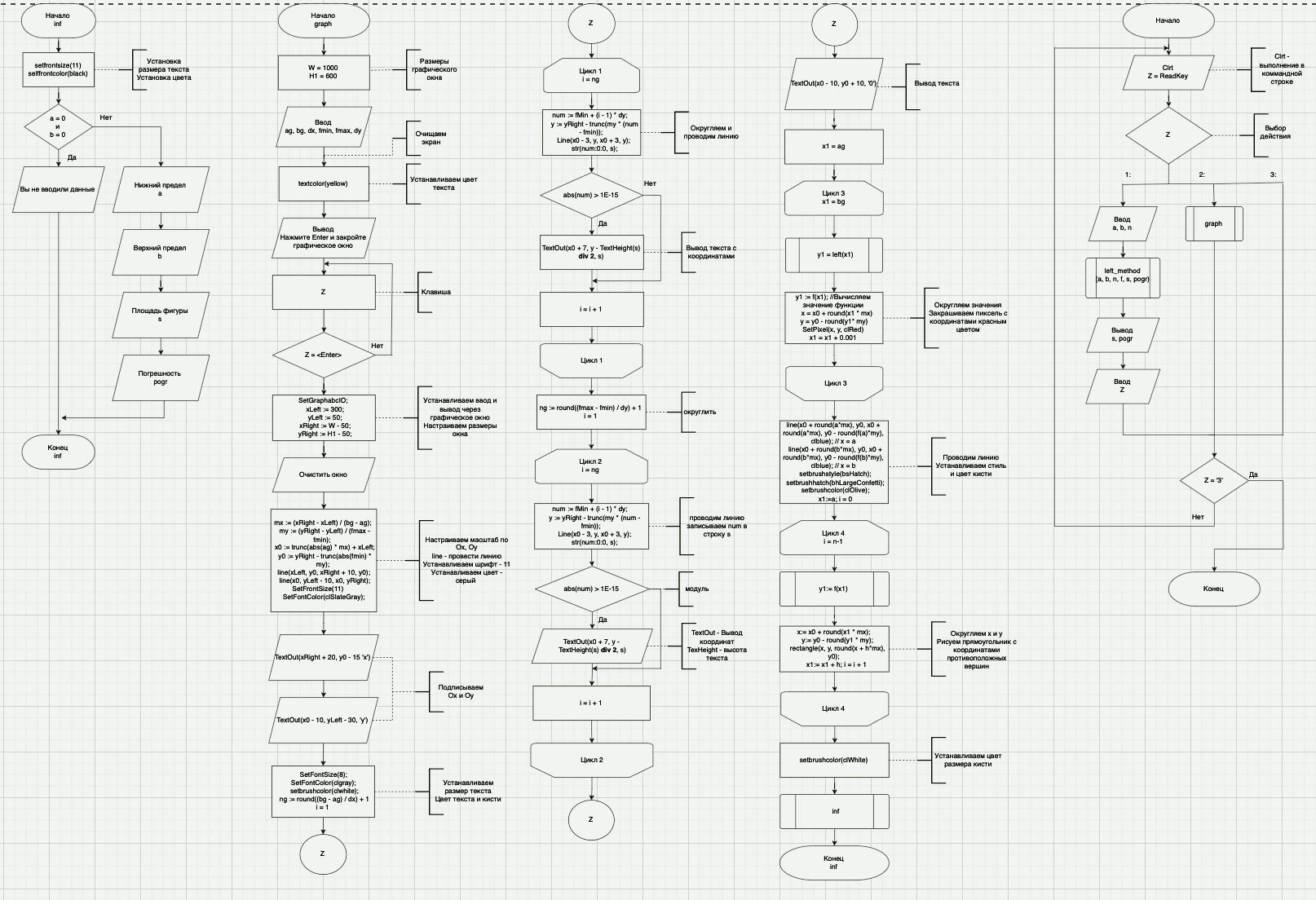
****

Рисунок 1 – Схема Алгоритма.

1. **Код программы:**
2. **Program** DKR4;
3. **uses** Crt, GraphABC;
4. **type**
5. bam = **function**(x: Real): Real;
6. **var**
7. a, b, dx, dy, h: Real;
8. n: Integer;
9. s, pogr: Real;
10. Z: char;
11. x, y: integer;
12. color : Byte;
13. **function** f(x: real): real;
14. **begin**
15. f:=1\*(x\*\*3)+2\*(x\*\*2)-2\*x+17;
16. **end**;
17. **procedure** left\_method(a, b: Real; n: Integer; func: bam; **var** s, pogr:real);
18. **var**
19. x: Real;
20. i: Integer;
21. fa, fb: real; //Значения первообразной
22. **begin**
23. h:= (b-a)/n;
24. x:= a;
25. **for** i:=0 **to** n-1 **do begin**
26. s:=s+f(x);
27. x:=x+h;
28. **end**;
29. s:=s\*h;
30. fa:=1/4\*a\*\*4+2/3\*a\*\*3-a\*\*2+17\*a; //нахождение значения первой первообразной через производную
31. fb:=1/4\*b\*\*4+2/3\*b\*\*3-b\*\*2+17\*b; //второй
32. pogr:=(fb-fa) - s; //нахождение пгрешности
33. **end**;
34. **procedure** inf;
35. **begin**
36. setfontsize(11);
37. setfontcolor(clblack);
38. **if not** ((a = 0) **and** (b = 0)) **then**
39. **begin**
40. writeln('Нижний предел: ', a);
41. writeln('Верхний предел: ', b);
42. writeln('Площадь заштрихованной фигуры: ', s:0:2);
43. writeln('Погрешность: ', pogr);
44. **end**
45. **else** writeln('Вы не вводили данные');
46. **end**;
47. **procedure** graph;
48. **const**
49. W = 1000; H1 = 600;//Размеры графического окна
50. **var**
51. x0, y0, x, y, xLeft, yLeft, xRight, yRight, ng: integer;
52. ag, bg, fmin, fmax, x1, y1, mx, my, num: real;
53. i: byte;
54. s: string;
55. **begin**
56. SetConsoleIO;
57. textcolor(11);
58. clrscr;
59. Writeln('Введите нижнюю границу системы координат по Х: ');
60. read(ag);
61. Writeln('Введите верхнюю границу системы координат по Х: ');
62. read(bg);
63. Writeln('Введите единичный отрезок по Х: ');
64. read(dx);
65. Writeln('Введите нижнюю границу системы координат по Y: ');
66. read(fmin);
67. Writeln('Введите верхнюю границу системы координат по Y: ');
68. read(fmax);
69. Writeln('Введите единичный отрезок по Y: ');
70. read(dy);
71. writeln;
72. clrscr;
73. textcolor(yellow);
74. Writeln('Нажмите [Enter] и откройте графическое окно');
75. **repeat**
76. Z := readkey;
77. **until** Z = #13;
78. SetGraphabcIO;
79. SetWindowSize(W, H1); //Устанавливаем размеры графического окна
80. xLeft := 300;
81. yLeft := 50;
82. xRight := W - 50;
83. yRight := H1 - 50;
84. clearwindow;
85. mx := (xRight - xLeft) / (bg - ag); //масштаб по Х
86. my := (yRight - yLeft) / (fmax - fmin); //масштаб по Y
87. x0 := trunc(abs(ag) \* mx) + xLeft;
88. y0 := yRight - trunc(abs(fmin) \* my);
89. line(xLeft, y0, xRight + 10, y0); //ось ОХ
90. line(x0, yLeft - 10, x0, yRight); //ось ОY
91. SetFontSize(11); //Размер шрифта
92. SetFontColor(clSlateGray); //Цвет шрифта
93. TextOut(xRight + 20, y0 - 15, 'х'); //Подписываем ось OX
94. TextOut(x0 - 10, yLeft - 30, 'у'); //Подписываем ось OY
95. SetFontSize(8); //Размер шрифта
96. SetFontColor(clgray); //Цвет шрифта
97. setbrushcolor(clwhite);
98. ng := round((bg - ag) / dx) + 1; //количество засечек по ОХ
99. **for** i := 1 **to** ng **do**
100. **begin**
101. num := ag + (i - 1) \* dx; //Координата на оси ОХ
102. x := xLeft + trunc(mx \* (num - ag)); //Координата num в окне
103. Line(x, y0 - 3, x, y0 + 3); //рисуем засечки на оси OX
104. str(Num:0:1, s);
105. **if** abs(num) > 1E-15 **then** //Исключаем 0 на оси OX
106. TextOut(x - TextWidth(s) **div** 2, y0 + 10, s)
107. **end**;
108. ng := round((fmax - fmin) / dy) + 1; //количество засечек по ОY
109. **for** i := 1 **to** ng **do**
110. **begin**
111. num := fMin + (i - 1) \* dy; //Координата на оси ОY
112. y := yRight - trunc(my \* (num - fmin));
113. Line(x0 - 3, y, x0 + 3, y); //рисуем засечки на оси Oy
114. str(num:0:0, s);
115. **if** abs(num) > 1E-15 **then** //Исключаем 0 на оси OY
116. TextOut(x0 + 7, y - TextHeight(s) **div** 2, s)
117. **end**;
118. TextOut(x0 - 10, y0 + 10, '0'); //Нулевая точка
119. x1 := ag; //Начальное значение аргумента
120. **while** x1 <= bg **do**
121. **begin**
122. y1 := f(x1); //Вычисляем значение функции
123. x := x0 + round(x1 \* mx); //Координата Х в графическом окне
124. y := y0 - round(y1 \* my); //Координата Y в графическом окне
125. SetPixel(x, y, clred);
126. x1 := x1 + 0.001 //Увеличиваем абсциссу
127. **end**;
128. line(x0 + round(a\*mx), y0, x0 + round(a\*mx), y0 - round(f(a)\*my), clblue); // х = а
129. line(x0 + round(b\*mx), y0, x0 + round(b\*mx), y0 - round(f(b)\*my), clblue); // х = b
130. setbrushstyle(bsHatch);
131. setbrushhatch(bhLargeConfetti);
132. setbrushcolor(clOlive);
133. x1:=a;
134. **for** i:=0 **to** n-1 **do**
135. **begin**
136. y1:= f(x1);
137. x:= x0 + round(x1 \* mx);
138. y:= y0 - round(y1 \* my);
139. rectangle(x, y, round(x + h\*mx), y0);
140. x1:= x1 + h;
141. **end**;
142. setbrushcolor(clWhite);
143. inf;
144. **end**;
145. **begin**
146. **repeat**
147. SetConsoleIO;
148. ClrScr;
149. textcolor(Yellow);
150. writeln('1. Вычисление площади фигуры, ограниченной кривой');
151. writeln('2. График');
152. writeln('3. Выход');
153. write('Выберите действие: ');
154. Z := ReadKey;
155. **case** Z **of**
156. '1':
157. **begin**
158. ClrScr;
159. Textcolor(Yellow);
160. writeln('Введите границы интегрирования: ');
161. readln(a, b);
162. Textcolor(Yellow);
163. writeln('Введите количество делений: ');
164. readln(n);
165. Left\_method(a, b, n, f, s, pogr);
166. Textcolor(LightGreen);
167. writeln('Площадь фигуры: ', s);
168. Textcolor(LightGreen);
169. writeln('Погрешность: ', pogr);
170. readln;
171. **end**;
172. '2': graph;
173. '3': halt;
174. **end**;
175. **until** Z = '3';
176. **end**.
177. **Результат выполнения программы:**

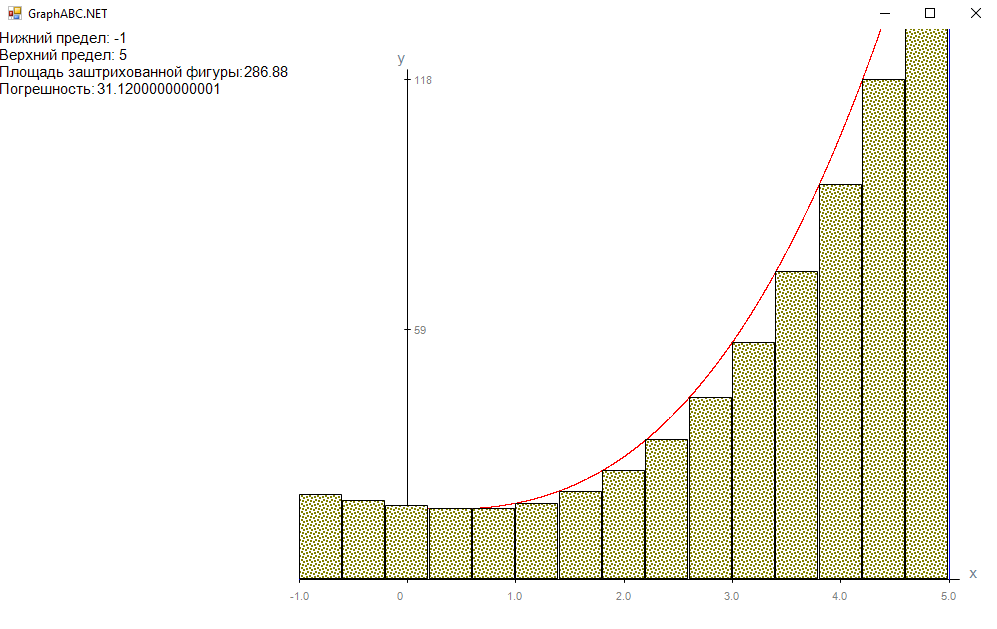
****

Рисунок 2 – Результат выполнения программы.

1. **Вывод.**

Данная работа прошла успешно. В работе было выполнено графическое отображение выполнение высчитывания площади криволинейной трапеции в системе координат.

В работе была написана программа по высчитыванию площади криволинейной трапеции по методу левых прямоугольников, графическое отображение на координатной плоскости, а также управление всеми процессами через кейс меню.

Трудности в работе возникли в построении схемы алгоритма и соединении связей в схеме алгоритма.

Таким образом, в данной работе были произведены операции с графическим модулем на основе языка программирования Paskal, нахождение площади криволинейной трапеции, и выполнение всех операций через кейс меню.