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-204-52-00

Максимов Антон Николаевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

1. Цель контрольной работы: освоить принципы работы в графическом режиме; получить базовые навыки взаимодействия с графическими примитивами.

2. Задание:

Вариант: 12.

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.

3. Схема алгоритма:

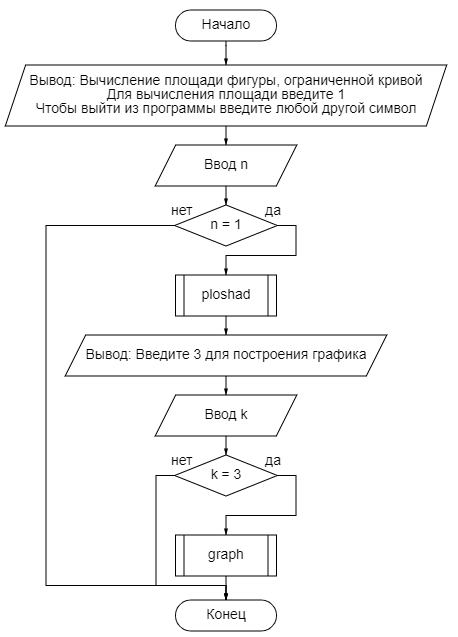


Рисунок 1 – Схема алгоритма (1/4)

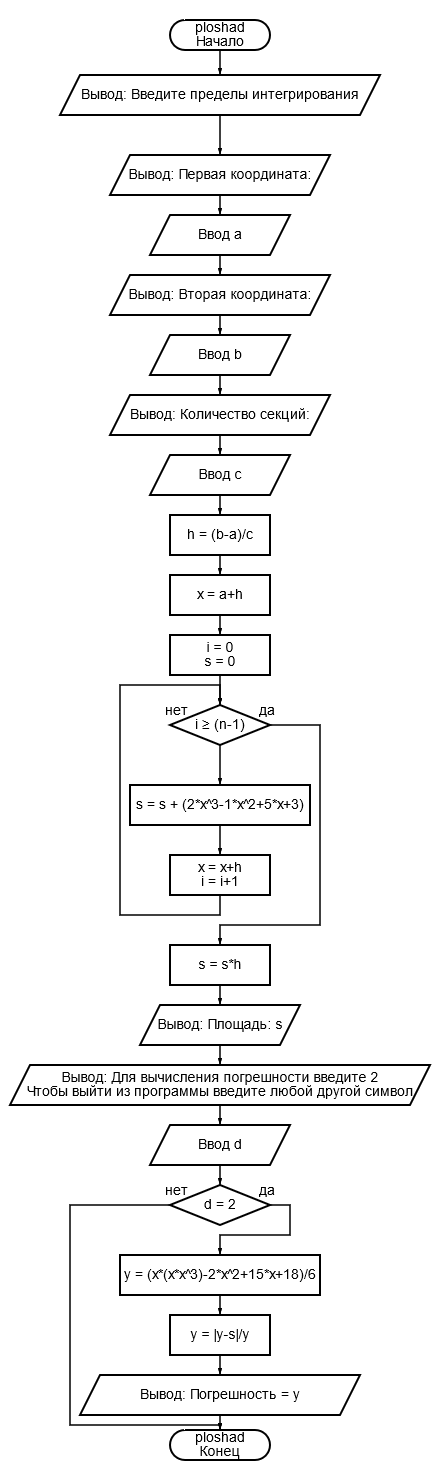


Рисунок 2 – Схема алгоритма (2/4)

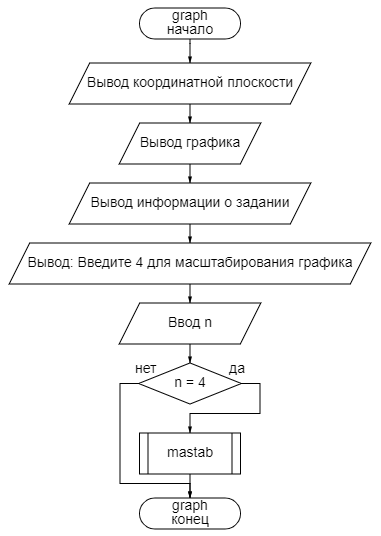


Рисунок 3 – Схема алгоритма (3/4)

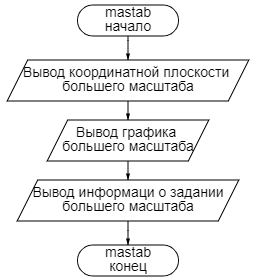


Рисунок 4 – Схема алгоритма (4/4)

4. Код программы:

**uses** crt;

**uses** graphABC;

**var** y: real; mashtX, mashtY, x0, y0, i1: integer;

**var** l: byte;

**function** vibor(**var** v: byte): byte;

**begin**

Textcolor(2);

writeln('Да - 1');

Textcolor(4);

writeln('Нет - 0');

Textcolor(15);

Write('Выберите вариант: ');

readln(v);

vibor := v;

**end**;

**var** l1: byte;

**function** viborp(**var** v: byte): byte;

**begin**

Textcolor(2);

textout(10, 130, 'Увеличить x - 1');

Textcolor(4);

Textcolor(2);

textout(10, 150, 'Увеличить y - 2');

Textcolor(4);

textout(10, 170, 'Уменьшить x - 3');

Textcolor(15);

textout(10, 190, 'Уменьшить y - 4');

Textcolor(15);

textout(10, 220, 'Закончить - 0');

textout(10, 250, 'Выберите вариант: ');

readln(v);

viborp := v;

**end**;

**var** n, i: integer; a, b, h, x, s, f, d: real;

**function** f0(**var** x: real): real;//основная

**begin**

**var** f1: real;

f1 := 1 \* power(x, 3) + (0) \* power(x, 2) + (5) \* x + (11);

f0 := f1;

**end**;

**function** f2(**var** x: real): real;//первообразная

**begin**

**var** f3: real;

f3 := 1 / 4 \* power(x, 4) + 5 / 2 \* power(x, 2) + 11 \* x;

f2 := f3;

**end**;

**var** m: integer;

**function** graph(**var** p, n1: integer; **var** s1, s2: real): real; //График

**var** i1, i, w, l, w1: integer;e: real;

**begin**

clearwindow;

MaximizeWindow;

x0 := windowwidth **div** 2;

y0 := windowheight **div** 2;

line(0, y0, windowwidth - 11, y0);

line(x0, 0, x0, windowheight - 11);

textout(x0 + 5, y0 + 5, '0');

textout(windowwidth - 10, y0 - 20, 'x');

textout(x0 + 5, 0, 'y');

textout(10, 10, 'Найти площадь фигуры ограниченной кривой 2x^3-1x^2+5x+3');

**for** i := 1 **to** 100 **do**

**begin**

line(x0 + round(i \* mashtX), y0 - 3, x0 + round(i \* mashtX), y0 + 3);

line(x0 - round(i \* mashtX), y0 - 3, x0 - round(i \* mashtX), y0 + 3);

line(x0 - 3, y0 + round(i \* mashtY), x0 + 3, y0 + round(i \* mashtY));

line(x0 - 3, y0 - round(i \* mashtY), x0 + 3, y0 - round(i \* mashtY));

textout(x0 + round(i \* mashtX), y0 + 10, inttostr(i));

textout(x0 - round(i \* mashtX), y0 + 5, inttostr(-i));

textout(x0 - 25, y0 - round(i \* mashtY), inttostr(i));

textout(x0 - 20, y0 + round(i \* mashtY), inttostr(-i));

**end**;

**for** i1 := -15000 **to** 15000 **do**

**begin**

x := 0.001 \* i1;

y := f0(x);

setpixel(x0 + round(mashtX \* x), y0 - round(mashtY \* y), clRed);

graph := p;

**end**;

setpencolor(clpurple);

line(x0 + round(s1 \* mashtX), 0, x0 + round(s1 \* mashtX), windowheight);

setpencolor(clorange);

line(x0 + round(s2 \* mashtX), 0, x0 + round(s2 \* mashtX), windowheight);

setpencolor(clblack);

l := round((x0 + round(s2 \* mashtX) - x0 - round(s1 \* mashtX)) / n1);

w := l;

w1 := 0;

e := x;

x := a;

**while** x <= b **do**

**begin**

setpixel(x0 + round(x \* mashtX), y0 - round(f0(x) \* mashtY), clred);

**if** (x0 + round(s1 \* mashtX)) = (x0 + round(x \* mashtX)) **then**

**begin**

setpencolor(clBlue);

e := x;

Rectangle(x0 + round(s1 \* mashtX), round(y0 - round(f0(e) \* mashtY)\*1.4), x0 + round(s1 \* mashtX + (w + l)), y0);

**end**;

**if** (x0 + round(s1 \* mashtX + l)) = (x0 + round(x \* mashtX)) **then**

**begin**

setpencolor(clBlue);

e := x;

**for var** v := x0 + round(s1 \* mashtX) **to** x0 + round(s2 \* mashtX) **do**

Rectangle(x0 + round(s1 \* mashtX + l), round(y0 - round(f0(e) \* mashtY)\*1.4), x0 + round(s1 \* mashtX + (w + l)), y0);

l := l + w;

e := e + l;

inc(w1);

**if** (w1 + 1) = n1 **then**

**begin**

setpencolor(clBlack);

**break**;

**end**;

**end**;

x := x + 0.001;

**end**;

**end**;

**function** mastab: integer;//Маштабирование

**begin**

textout(10, 100, 'Увеличить масштаб графика?');

viborp(l1);

**case** l1 **of**

1:

**begin**

mashtX := mashtX + 10;

graph(m, n, a, b);

mastab;

**end**;

2:

**begin**

mashtY := mashtY + 10;

graph(m, n, a, b);

mastab;

**end**;

3:

**begin**

mashtY := mashtX - 10;

graph(m, n, a, b);

mastab;

**end**;

4:

**begin**

mashtY := mashtY - 10;

graph(m, n, a, b);

mastab;

**end**;

0:

**begin**

writeln('Завершение программы...');

**exit**();

**end**;

**end**;

mastab := 0

**end**;

**begin**

textcolor(11);

writeln('Найти площадь фигуры ограниченной кривой 2x^3-1x^2+5x+3');

textcolor(15);

writeln('Введите пределы интегрирования a и b (a<b)');

textcolor(4);

readln(a, b);

**if** (a > b) **then begin**

textcolor(14); writeln('a должен быть меньше b'); **exit**(); **end**;

textcolor(15);

writeln('Введите количество секций');

textcolor(3);

readln(n);

textcolor(15);

h := (b - a) / n;

x := a + h;

**for** i := 0 **to** n - 1 **do**

**begin**

f := f0(x);

S := S + f;

x := x + h;

**end**;

S := (h \* (f0(a) + f0(b))) / 2 + S;

textcolor(14);

writeln('Площадь = ', s);

textcolor(15);

writeln('Вывести погрешность?');

vibor(l);

mashtX := 20;

mashtY := 20;

**case** l **of**

1:

**begin**

textcolor(2);

d := abs((f2(b) - f2(a)) - S);

writeln('Погрешность = ', d);

textcolor(15);

**end**;

0:

**end**;

writeln('Вывести график функции?');

vibor(l);

**case** l **of**

1:

**begin**

graph(m, n, a, b);

**end**;

0:

**begin**

writeln('Завершение программы...');

**exit**();

**end**;

**end**;

mastab;

**end**.

5. Результат выполнения программы:

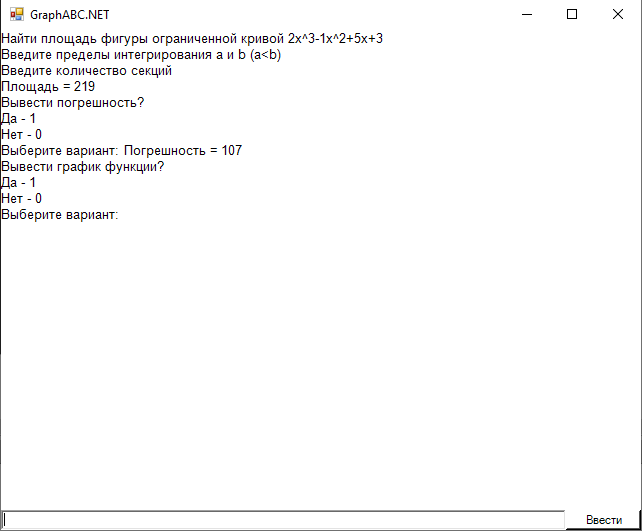


Рисунок 3 – Результат выполнения программы (1/2)

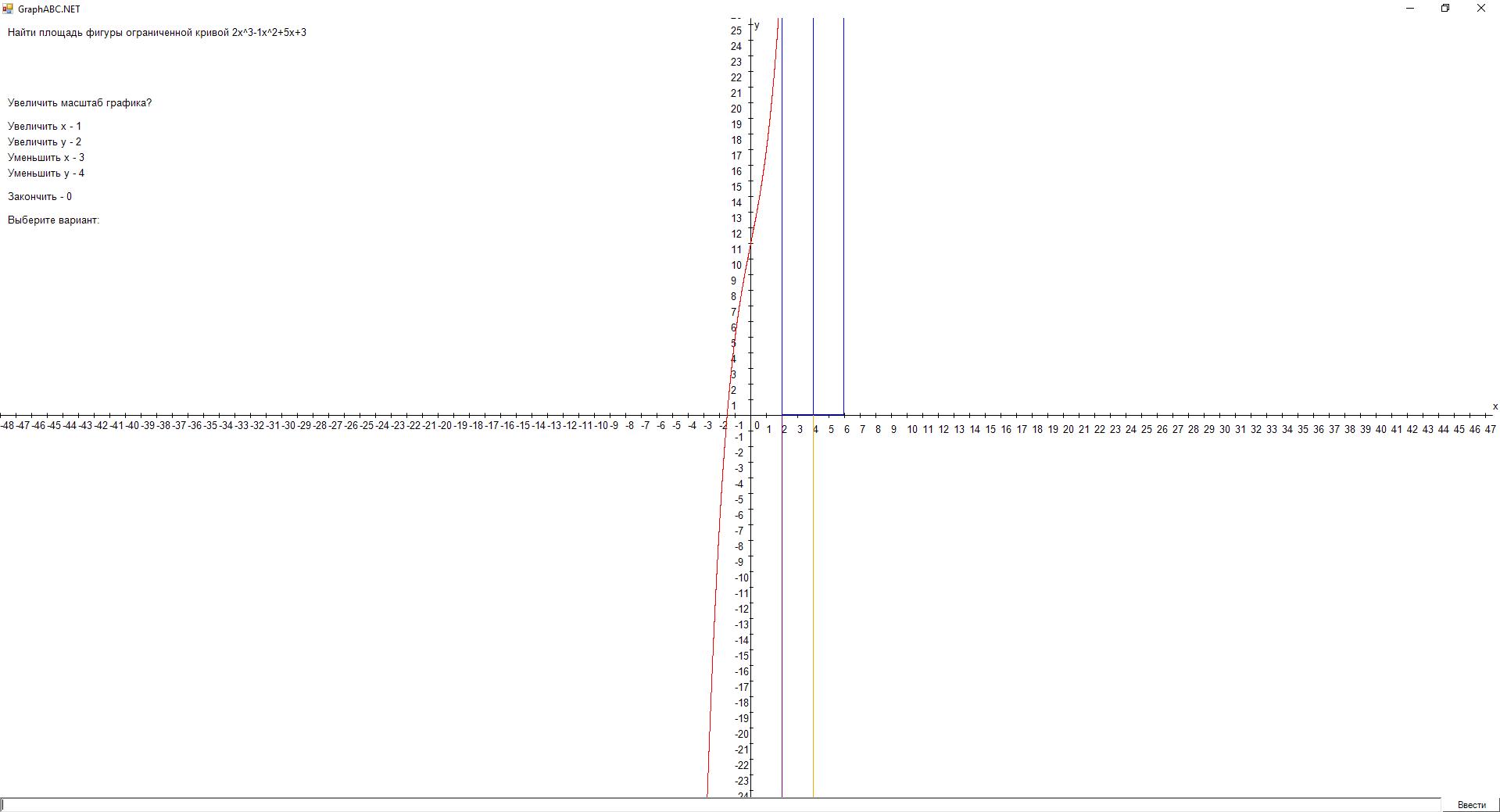


Рисунок 4 – Результат выполнения программы (2/2)

1. Вывод:

Для выполнения задания были использованы функции библиотеки GraphABC языка Pascal.

Для построения координатной плоскости, вывода графика, масштабирования графика, штриховки фигуры, ограничиваемой графиком и вывода подписей были использованы следующие функции: line, moveto, lineto, textout, setpencolor, setfontsize, setfontcolor.

Также были применены функции библиотеки CRT для организации взаимодействия с пользователем посредством case-меню.

К вышеперечисленному можно добавить использование операторов if, циклов for, внутриблочное описание переменных, инициализации переменных при описании, расширенное присваивание, для вывода используется функция Print.

Благодаря всему вышеперечисленному задание было успешно выполнено.