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 АЛГОРИДМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-204-52-00

Утёмова Ксения Александровна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

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

**Формулировка задания:**

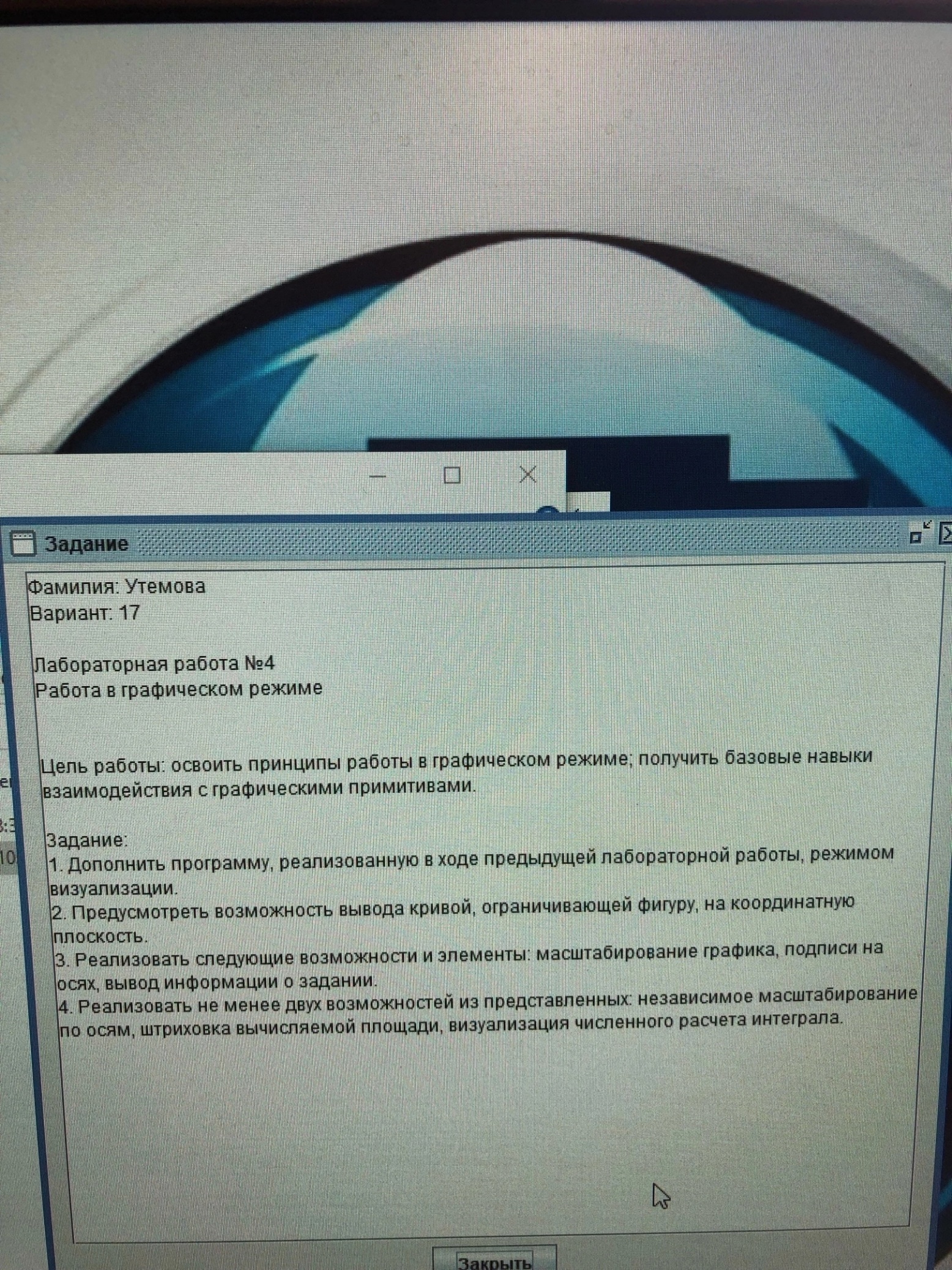
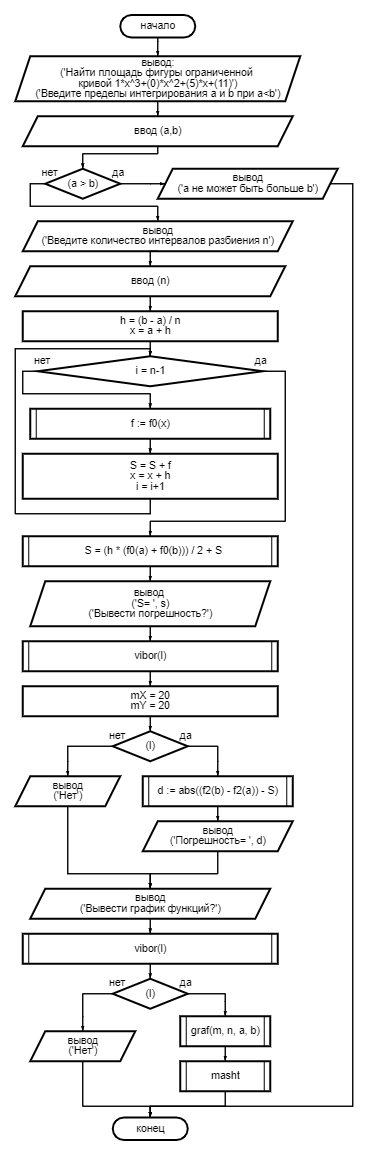


Рисунок 1 - Скриншот задания.

**Схемы алгоритмов с комментариями:**

1. **Алгоритм для всей задачи:**



-1 2 6



0

1



1

0



Рисунок 2 - Схема алгоритма для всей задачи.

1. **Схемы алгоритмов для функций:**

Изображение выглядит как текст, диаграмма, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 - Алгоритм для функции (vibor)

Изображение выглядит как текст, диаграмма, зарисовка, Шрифт

Автоматически созданное описание

Рисунок 4 - Алгоритм функции (f0)

Изображение выглядит как текст, диаграмма, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 5 - Алгоритм функции (f2)

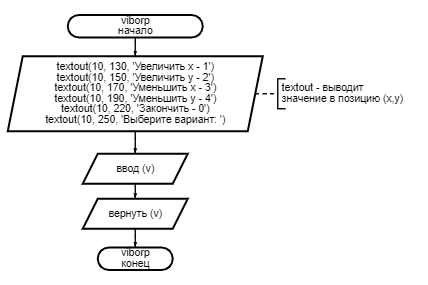
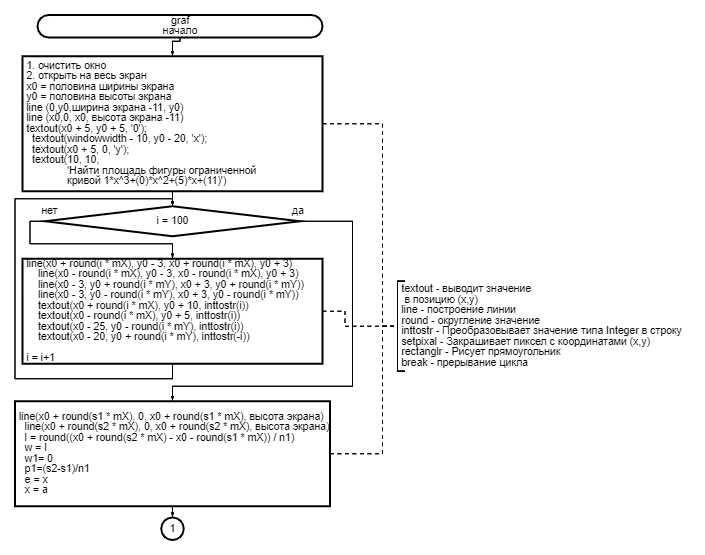


Рисунок 6 - Алгоритм функции (viborp)



-1

6

2



Рисунок 7 - Алгоритм функции (graf)

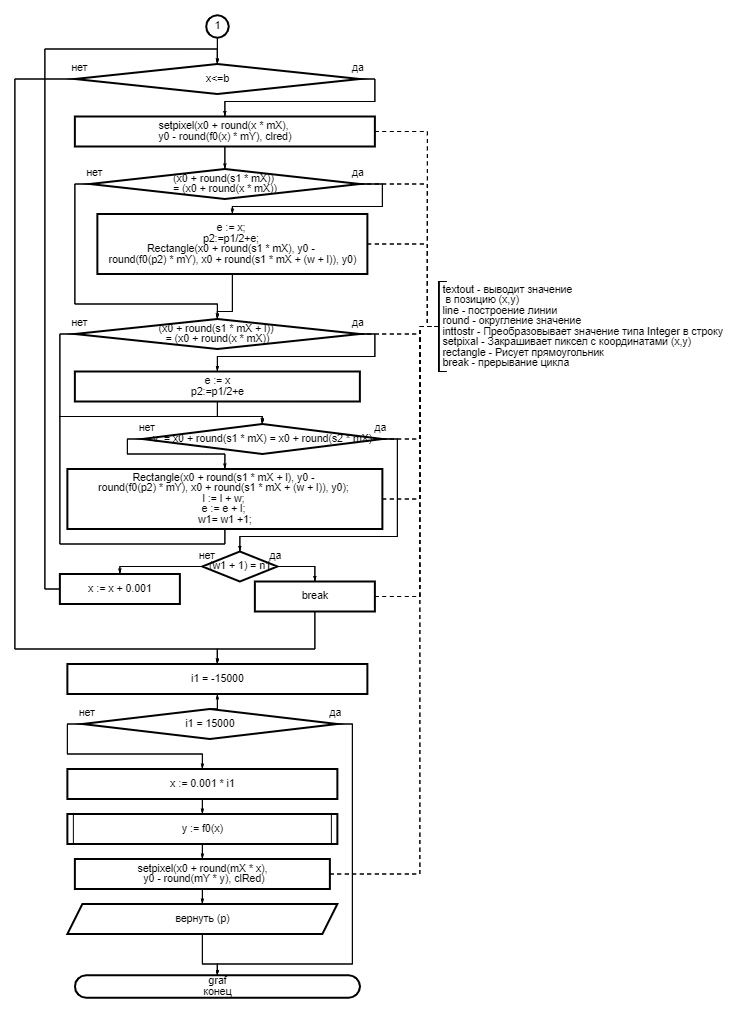
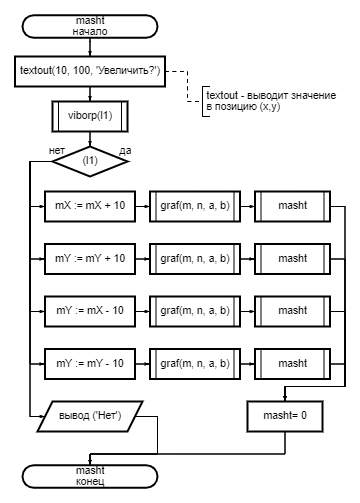


Рисунок 8 – Продолжение алгоритма для функции (graf)



0

4

3

2

1



Рисунок 9 - Алгоритм функции (masht)

**Код программы:**

**uses** graphABC;

**var** y: real; mX,mY, x0, y0, i1: integer;

l: byte;

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

**begin**

writeln('Да - 1');

writeln('Нет - 0');

Write('Выберите вариант: ');

readln(v);

vibor := v;

**end**;

**var** l1: byte;

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

**begin**

textout(10, 130, 'Увеличить x - 1');

textout(10, 150, 'Увеличить y - 2');

textout(10, 170, 'Уменьшить x - 3');

textout(10, 190, 'Уменьшить y - 4');

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)+(-1)\*power(x,2)+(2)\*x+(6);

f0 := f1;

**end**;

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

**begin**

**var** f3: real;

f3 := (power(x, 4)/4) - (power(x, 3)/3) +(power(x, 2))+(6\*x);

f2 := f3;

**end**;

**var**

m: integer;

**function** graf(**var** p, n1: integer; **var** s1, s2: real): real; //Вывод графика

**var**

i1, i, w, l, w1: integer; p1,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 1\*x^3+(-1)\*x^2+(2)\*x+(6)');

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

**begin**

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

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

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

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

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

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

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

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

**end**;

setpencolor(clblue);

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

setpencolor(clgreen);

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

setpencolor(clblack);

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

w := l;

w1 := 0;

p1:=(s2-s1)/n1;

e := x;

x := a;

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

**begin**

setpixel(x0 + round(x \* mX), y0 - round(f0(x) \* mY), clred);

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

**begin**

setpencolor(clRed);

e := x;

**var** p2:=p1/2+e;

Rectangle(x0 + round(s1 \* mX), y0 - round(f0(p2) \* mY), x0 + round(s1 \* mX + (w + l)), y0);

**end**;

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

**begin**

setpencolor(clRed);

e := x;

**var** p2:=p1/2+e;

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

Rectangle(x0 + round(s1 \* mX + l), y0 - round(f0(p2) \* mY), x0 + round(s1 \* mX + (w + l)), y0);

l := l + w;

e := e + l;

w1:= w1 +1;

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

**begin**

setpencolor(clBlack);

**break**;

**end**;

**end**;

x := x + 0.001;

**end**;

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

**begin**

x := 0.001 \* i1;

y := f0(x);

setpixel(x0 + round(mX \* x), y0 - round(mY \* y), clRed);

graf := p;

**end**;

**end**;

**function** masht: integer;//Маштабирование

**begin**

textout(10, 100, 'Увеличить?');

viborp(l1);

**case** l1 **of**

1:

**begin**

mX := mX + 10;

graf(m, n, a, b);

masht;

**end**;

2:

**begin**

mY := mY + 10;

graf(m, n, a, b);

masht;

**end**;

3:

**begin**

mY := mX - 10;

graf(m, n, a, b);

masht;

**end**;

4:

**begin**

mY := mY - 10;

graf(m, n, a, b);

masht;

**end**;

0:

**begin**

writeln('Нет');

**exit**();

**end**;

**end**;

masht:= 0

**end**;

**begin**

writeln('Найти площадь фигуры ограниченной кривой 1\*x^3+(-1)\*x^2+(2)\*x+(6)');

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

readln(a, b);

**if** (a > b) **then begin** writeln('a не может быть больше b'); **exit**(); **end**;

writeln('Введите количество интервалов разбиения n');

readln(n);

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;

writeln('S= ', s);

writeln('Вывести погрешность?');

vibor(l);

mX := 20;

mY := 20;

**case** l **of**

1:

**begin**

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

writeln('Погрешность= ', d);

**end**;

0:

**begin**

writeln('Нет');

**end**;

**end**;

writeln('Вывести график функций?');

vibor(l);

**case** l **of**

1:

**begin**

graf(m, n, a, b);

**end**;

0:

**begin**

writeln('Нет');

**exit**();

**end**;

**end**;

masht;

**end**.

**Результаты выполнения программ:**

1. **Полное выполнение задания.**

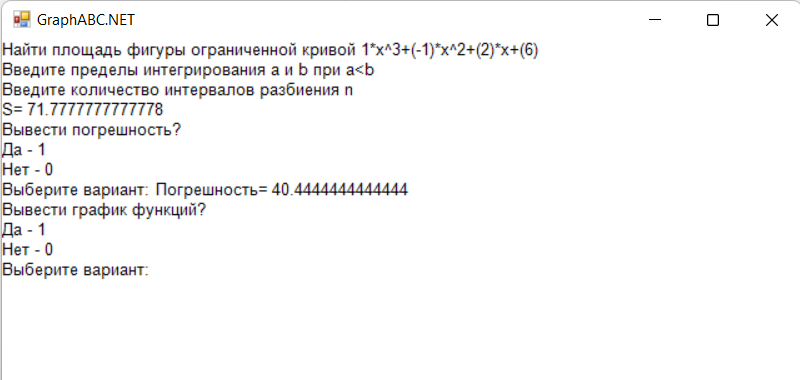


Рисунок 10 - Результат №1 с выводом погрешности.

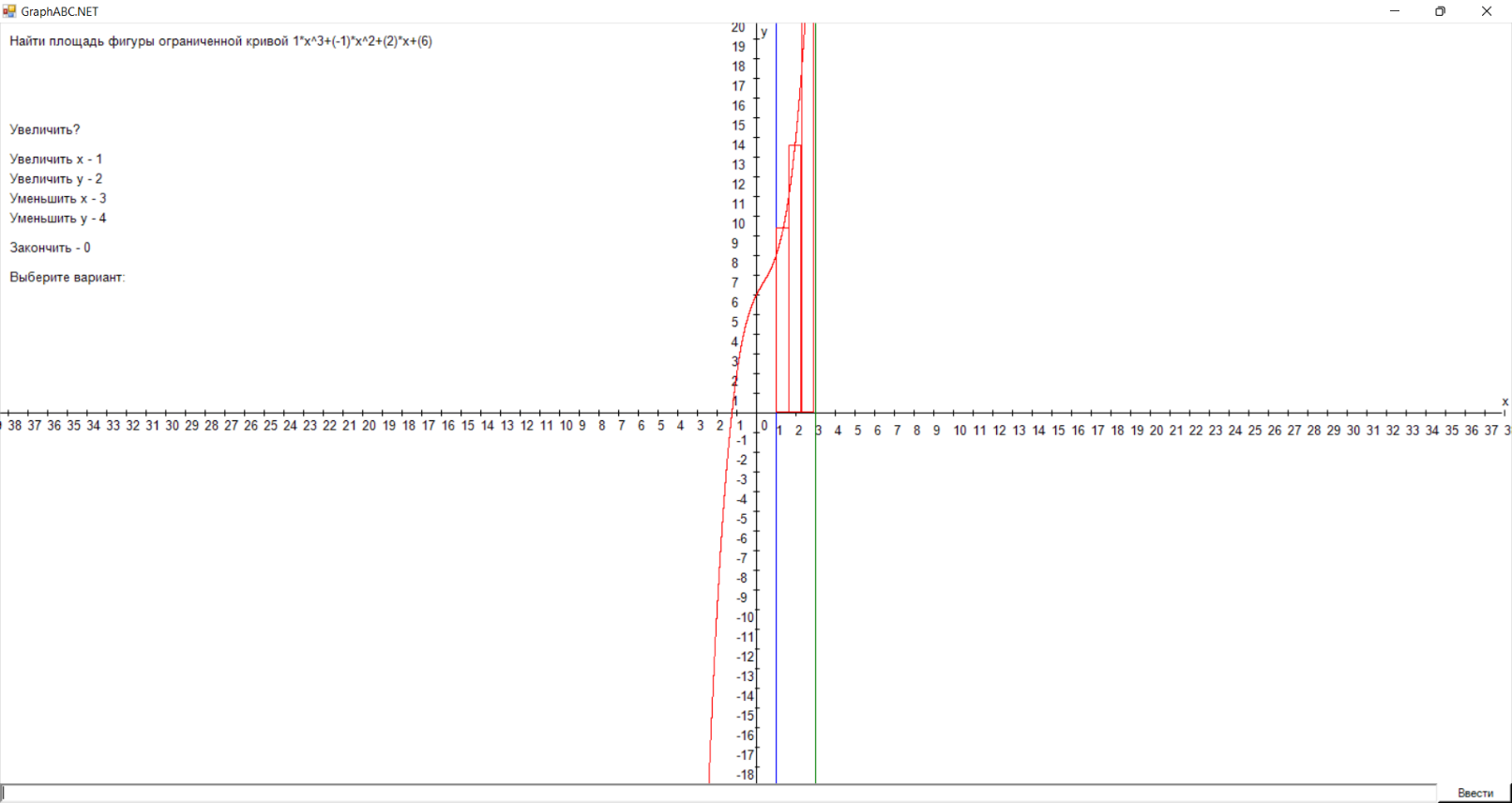


Рисунок 11 - Результат №1 с выводом графика.

1. **Результат при изменении масштаба для графика:**

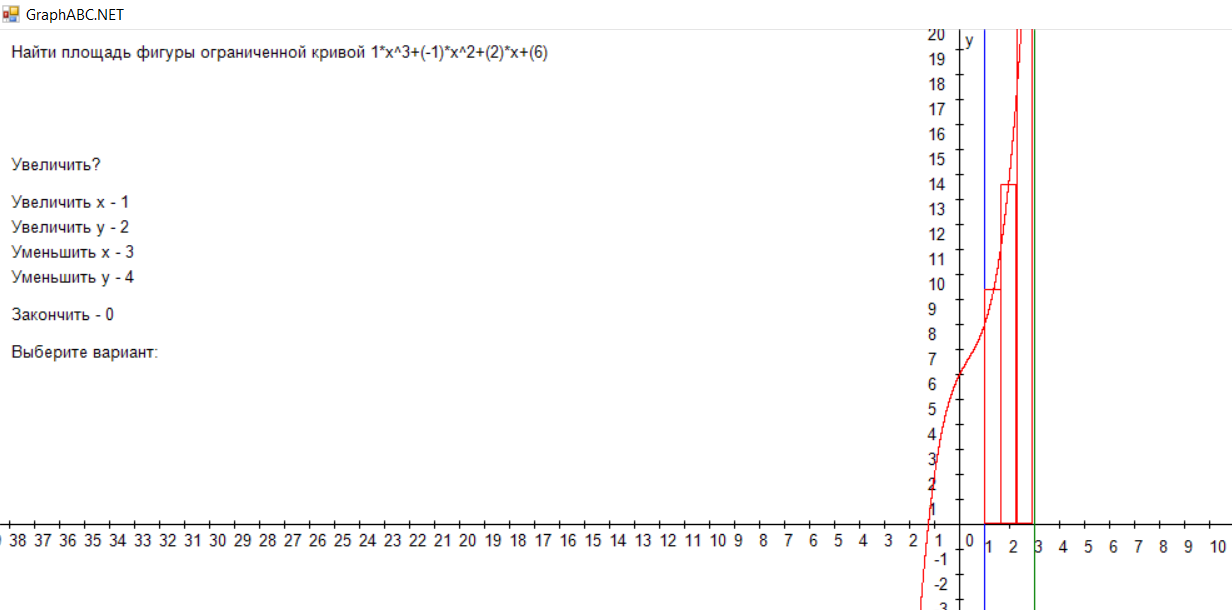


Рисунок 12 – Результат при начальном выводе графика.

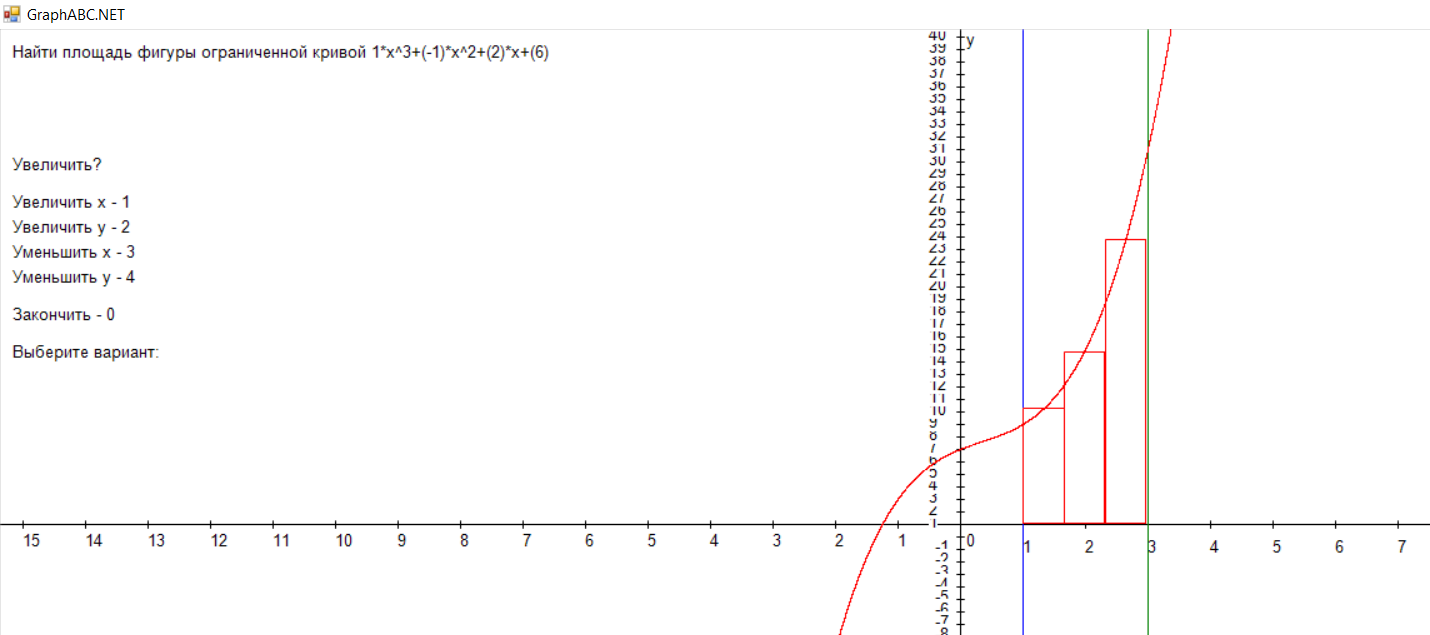


Рисунок 13 - Результат при изменении масштаба для графика.

1. **Вывод ошибки о введенных данных**.

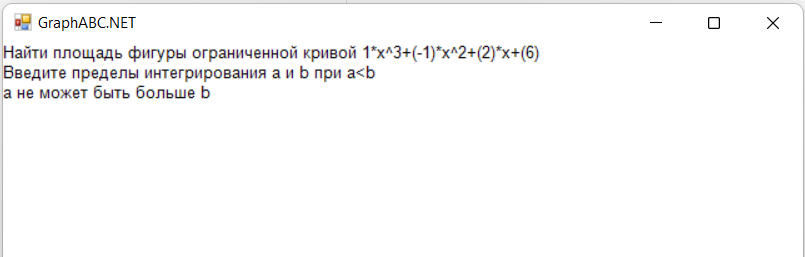
****

Рисунок 14 - Результат №2.

Результаты программ мы получили правильные, соответствующие ожидаемым ответам. Полностью выполняются все условия задания. Все рисунки и надписи выводятся верно, по заданных в программе координатам.

**Вывод:**

Во время выполнения четвертой домашней контрольной работы мы смогли написать программные коды для решения поставленных задач по теме “Работа в графическом режиме.” В этой работе было изучено построение графика функций в графическом режиме. С помощью созданной программы получилось вычислить площадь фигуры, ограниченной кривой 1\*x^3+(-1) \*x^2+(2) \*x+(6), использовав метод средних прямоугольников.

Результатом нашей работы стало решение поставленной задачи и вывод на экран графических элементов, с помощью использование, встроенного в программу модуля GraphABC, на языке программирования Pascal, в программе PascalABC.Net.

Смогли освоить принципы работы в графическом режиме; получить базовые навыки взаимодействия с графическими примитивами. Закрепили умение создавать схемы алгоритма и разобрались, как писать и оформлять отчеты по учебной дисциплине “Основы алгоритмизации и программирования.”